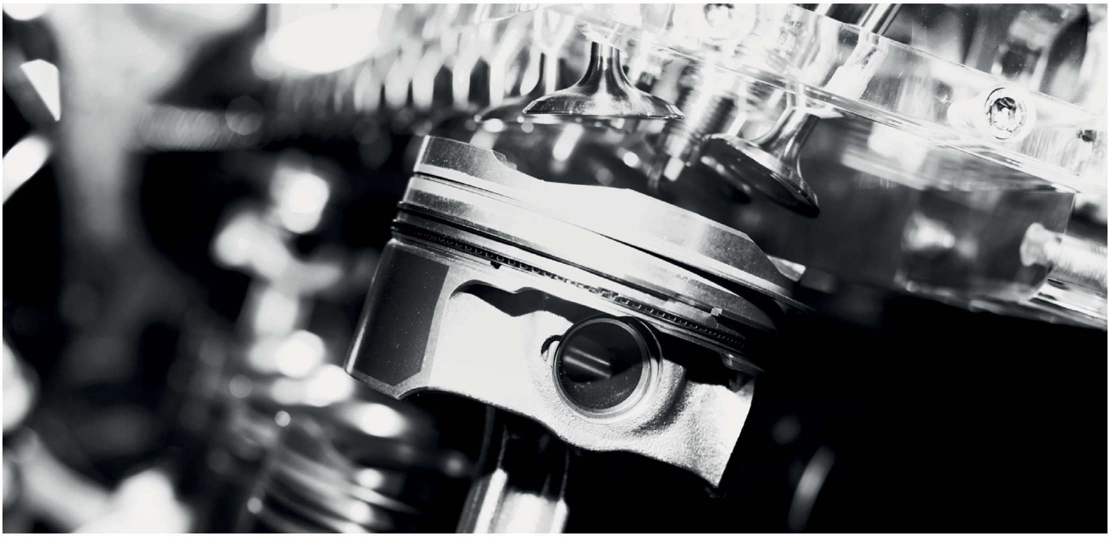
Advances in Automation Engineering Band 10

Editor: Clemens Gühmann

Nino Sandmeier

Optimization of adaptive test design methods for the determination of steady-state data-driven models in terms of combustion engine calibration



Universitätsverlag der TU Berlin

Technische Universität Berlin

Nino Sandmeier

Optimization of adaptive test design methods for the determination of steady-state data-driven models in terms of combustion engine calibration

The scientific series *Advances in Automation Engineering* is edited by Prof. Dr.-Ing. Clemens Gühmann.

Advances in Automation Engineering | 10

Nino Sandmeier

Optimization of adaptive test design methods for the determination of steady-state data-driven models in terms of combustion engine calibration

Universitätsverlag der TU Berlin

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the internet at http://dnb.dnb.de.

Universitätsverlag der TU Berlin, 2022
https://verlag.tu-berlin.de

Fasanenstr. 88, 10623 Berlin
Tel.: +49 (0)30 314 76131
E-Mail: publikationen@ub.tu-berlin.de

Zugl.: Berlin, Techn. Univ., Diss., 2021
Gutachter: Prof. Dr.-Ing. Clemens Gühmann
Gutachter: Prof. Dr.-Ing. Hermann Rottengruber
Gutachter: Prof. Dr.-Ing. Hanno Ihme-Schramm
Die Arbeit wurde am 14. Dezember 2021 an der Fakultät IV unter Vorsitz von Prof. Dr.-Ing. Julia Kowal erfolgreich verteidigt.

Cover image:
https://pixabay.com/de/photos/motor-kolben-maschine-nahaufnahme-2154575
by andreas160578 | Pixabay License (https://pixabay.com/de/service/license/)

Print: docupoint GmbH
Layout/Typesetting: Nino Sandmeier

ORCID iD Nino Sandmeier: 0000-0003-2245-2473
https://orcid.org/0000-0003-2245-2473

ISBN 978-3-7983-3247-8 (print)
ISBN 978-3-7983-3248-5 (online)

ISSN 2509-8950 (print)
ISSN 2509-8969 (online)

Published online on the institutional repository of the Technische Universität Berlin:
DOI 10.14279/depositonce-15364
http://dx.doi.org/10.14279/depositonce-15364

Acknowledgment

This dissertation was created during my time at the IAV GmbH in Berlin, Germany. The close cooperation between IAV and the Technical University of Berlin made this research possible. I would like to thank both organizations for giving me this opportunity.

This research was supervised by Prof. Dr.-Ing. Clemens Gühmann, head of the chair Electronic Measurement and Diagnostic Technology of the Institute of Energy and Automation Technology. I thank him for his mentoring, in particular for his interest and the deep technical insight into the topic and the consequentially very helpful advice. At IAV Berlin, Dr.-Ing. Karsten Röpke was my supervisor during the time of creation. I would like to thank him very much for the possibility to investigate this topic, for the long-lasting support, and the necessary leeway for realizing this research. I would also like to thank Steffen Schaum for always backing me, enabling my personal development during the research time, and being the driver of this research topic at IAV. I am particularly grateful for the assistance given by my colleagues Dr.-habil. Michael Hegmann and Florian Seele. Several, long-lasting discussions with Michael Hegmann directed the research and enabled the outcome as it is present in this thesis. During my time at IAV, Florian Seele was always supporting me, gave me a broad insight into the software development topic and enabled the software implementation, which was necessary to apply the developed algorithms. Besides the named colleagues, I would like to thank all other employees of the department of Karsten Röpke and other involved departments for the kind atmosphere, all technical discussions, and their participation in my personal and professional development.

I would like to thank Prof. Dr.-Ing. Hermann Rottengruber for his interest in the research topic and for being the second examiner. Special thanks should be given to my mentor and third examiner Prof. Dr.-Ing. Hanno Ihme-Schramm. Over the last years he always accompanied me,

starting as my lector during the master's program. Despite the distance, he always sought to provide a great support over several years. His technical advice, and especially the human factor point of view, shaped the content of this research and my personal perspectives. For chairing the thesis committee I would like to thank Prof. Dr.-Ing. Julia Kowal.

I am very grateful to my parents for their enduring support during my entire life, my education, personal development, and during this research. Without this support, I would not be where I am today.

Abstract

This thesis deals with the development of a model-based adaptive test design strategy with a focus on steady-state combustion engine calibration. The first research topic investigates the question how to handle limits in the input domain during an adaptive test design procedure. The second area of scope aims at identifying the test design method providing the best model quality improvement in terms of overall model prediction error.

To consider restricted areas in the input domain, a convex hull-based solution involving a convex cone algorithm is developed, the outcome of which serves as a boundary model for a test point search. A solution is derived to enable the application of the boundary model to high-dimensional problems without calculating the exact convex hull and cones. Furthermore, different data-driven engine modeling methods are compared, resulting in the Gaussian process model as the most suitable one for a model-based calibration. To determine an appropriate test design method for a Gaussian process model application, two new strategies are developed and compared to state-of-the-art methods. A simulation-based study shows the most benefit applying a modified mutual information test design, followed by a newly developed relevance-based test design with less computational effort. The boundary model and the relevance-based test design are integrated into a multicriterial test design strategy that is tailored to match the requirements of combustion engine test bench measurements. A simulation-based study with seven and nine input parameters and four outputs each offered an average model quality improvement of 36 % and an average measured input area volume increase of 65 % compared to a non-adaptive space-filling test design. The multicriterial test design was applied to a test bench measurement with seven inputs for verification. Compared to a space-filling test design measurement, the improvement could be confirmed with an average model quality increase of 17 % over eight outputs and a 34 % larger measured input area.

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Entwicklung einer modellbasierten adaptiven Versuchsplanungsstrategie für die Anwendung in der Applikation des Stationärverhaltens von Verbrennungsmotoren. Der erste Forschungsteil untersucht, wie sich Grenzen im Eingangsraum in die Versuchsplanung eines adaptiven Prozesses einbinden lassen. Ein weiterer Fokus liegt auf der Identifikation einer modellbasierten Versuchsplanung, die eine bestmögliche Verbesserung der globalen Modellqualität hinsichtlich des Prädiktionsfehlers ermöglicht.

Es wird ein Grenzraummodell auf Basis der konvexen Hülle unter Zuhilfenahme eines Algorithmus zur Bestimmung eines konvexen Konus entwickelt, das als Grundlage für eine Versuchsplanung in beschränkten Eingangsräumen verwendet wird. Um die Anwendbarkeit bei hochdimensionalen Problemstellungen zu gewährleisten, wird ein Verfahren vorgestellt, das eine Berechnung auch ohne die Bestimmung der exakten konvexen Hülle und konvexen Konen ermöglicht. Des Weiteren werden verschiedene Methoden zur datengetriebenen Modellbildung des Verbrennungsmotors verglichen, wobei das Gauß-Prozess Modell als die geeignetste Modellierungsmethode hervorgeht. Um die bestmögliche Versuchsplanungsmethode bei der Anwendung des Gauß-Prozess Modells zu ermitteln, werden zwei neue Strategien entwickelt und mit verfügbaren Methoden aus der Literatur verglichen. Eine simulationsbasierte Studie zeigt, dass eine angepasste Mutual Information Methode die besten Ergebnisse liefert. Ein neu entwickeltes relevanzbasiertes Verfahren erreicht die zweitbesten Ergebnisse, bietet aber einen geringeren Berechnungsaufwand als das Mutual Information Verfahren. Das Grenzmodell und das relevanzbasierte Verfahren werden in einem multikriteriellen Versuchsplanungsverfahren zusammengeführt, das an die Anforderungen von Messungen an einem Verbrennungsmotorenprüfstand angepasst ist. In einer simulationsbasierten Studie mit sieben bzw. neun Eingangsparametern und jeweils vier Ausgängen konnte eine durchschnittliche Modellqualitätsverbesserung von 36 % und eine mittlere Vergrößerung des vermessenen Eingangsraumvolumens von 65 % im

Vergleich zu einer nichtadaptiven raumfüllenden Versuchsplanung gezeigt werden. Das multikriterielle Versuchsplanungsverfahren wurde anhand von Prüfstandsmessungen mit sieben Eingangsparametern verifiziert. Im Vergleich zu einer raumfüllenden Versuchsplanung konnte eine mittlere Modellqualitätsverbesserung über alle acht Ausgänge von 17 % und ein um 34 % vergrößertes vermessenes Eingangsraumvolumen erreicht werden, wodurch die Ergebnisse der Simulationen bestätigt werden konnten.

Contents

Acronyms

AUC	area under the curve
BSFC	brake specific fuel consumption
CA	crank angle
CCD	central composite design
CO	carbon monoxide
COR	custom output range
COV	coefficient of variation
DoE	design of experiments
ECU	engine control unit
EGR	exhaust gas recirculation
GJK	Gilbert-Johnson-Keerthi algorithm
GPM	Gaussian process model
HILOMOT	hierarchical local model tree
IMSE	integrated mean squared error

LHD	Latin hypercube design
LOLIMOT	local linear model tree
LOOCV	leave one out cross-validation
MAP	maximum a posteriori
MBC	model-based calibration
MFB50	50% mass fraction burned
MI	mutual information
ML	maximum likelihood
MLP	multilayer perceptron
MMSE	maximum mean squared error
MS	soot mass
MSE	mean squared error
PEV	prediction error variance
PN	particulate number
QBC	query by committee
RBF	radial basis function
RHD	rapid hull determination
RMSE	root mean squared error
SE	squared exponential
SVM	support vector machine
TCP	transport control protocol
WG	waste gate position

Symbols

$\boldsymbol{A}$	normal vector matrix
B	test design exclusion area
$\boldsymbol{b}$	Hesse normal form distance vector
$\boldsymbol{\beta}$	model parameters

$\boldsymbol{c}$	basis function center
$\mathrm{co}(\boldsymbol{X})$	convex hull of $\boldsymbol{X}$
$\mathrm{cone}(\boldsymbol{X})$	convex cone given $\boldsymbol{X}$
d	Euclidean distance
Δ	domain width
δ_{ij}	kronecker delta
$\dim$	dimension
ϵ	noise
$\boldsymbol{\gamma}$	convex hull center point
$\mathcal{I}(\boldsymbol{x}^*)$	mutual information of potential test point $\boldsymbol{x}^*$
k_σ	smoothness factor
$\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$	covariance matrix given $\boldsymbol{X}$
$k(\boldsymbol{x}, \boldsymbol{x}')$	covariance function
l^2	Gaussian process length scale hyperparameter
λ	regularization parameter
$\boldsymbol{M}$	design matrix
$\dot{m}_{\text{fuel}}$	fuel mass flow
$\dot{m}_{\text{Soot}}$	soot mass flow
μ	mean value
$\mathcal{N}$	Gaussian distribution
ν_w	worst normalized error
$\boldsymbol{o}$	boundary labels
$\boldsymbol{p}$	split direction parameter vector
Φ	basis function
$\boldsymbol{R}$	covariance matrix of the parameters
r	validation RMSE
ρ	support value
$\boldsymbol{S}$	candidate set
$\boldsymbol{\Sigma}$	covariance matrix

σ	variance
τ	membership function
$\mathrm{T}_{\mathrm{Bef\,Turbine}}$	exhaust temperature before turbine
$\boldsymbol{\theta}$	hyperparameter vector
$\boldsymbol{v}$	model variance vector
$\boldsymbol{v}_{sn}$	support vector
W	Wiener process
$\boldsymbol{X}$	measurement matrix
$\boldsymbol{x}'$	next test point
$\boldsymbol{x}_*$	prediction input vector
y	process output
ζ	validity function

1 Introduction

The optimization of a technical system, and in the scope of this thesis the optimization of an internal combustion engine, relies on different approaches to find an optimal solution within the shortest possible time. In the past several methods were investigated to find solutions that satisfy different challenging requirements. To understand both the requirements as well as the method that is addressed in this thesis, the following will discuss the scope of application and the differentiation of the adaptive test design method to other existing methods.

1.1 Motivation

The process of combustion engine calibration is defined as the adjustment of maps and curves that are part of the engine control unit (ECU) software. Among other necessary tasks, the functions inside the ECU calculate the desired values for all actuators to operate the engine in each situation. By changing the content of the maps and curves, which is the task of a calibration engineer, the actuator behavior is influenced. Two conflicting requirements are the main drivers of combustion engine calibration. On the one hand, there are legislative requirements that restrict the combustion exhaust gas composition. Over the years the limits for the different exhausted gases are restricted further, where the reduction of hazardous emissions is in conflict to the reduction of environmental gases like carbon dioxide. On the other hand, the customer demands larger, more comfortable vehicles with an increasing amount of driver assistant systems. These demands have in common that they could lead to an increase of vehicle weight and its driving resistance in case no counteractive measures are introduced, which could result in a higher fuel consumption and therefore higher carbon dioxide emission. Next to the application of lightweight structures, another possibility to overcome these conflicting requirements is the development of more efficient engines that unfortunately increase the calibration complexity due to a raising amount of actuators. Quintuple injection [Hei+13] or a variable compression ratio combined with a turbocharger and a variable intake and exhaust camshaft in gasoline engines [KMK17; Nis20] are only an excerpt of today's complex technical solutions in mass production vehicles to lower the fuel consumption and meet the legislative requirements. Modern diesel engines offer up to 20 actuator parameters to find an optimal solution for. With each additional actuator to be calibrated, the need of steady-state measurement increases exponentially due to the curse of dimensionality that is firstly described in [Bel61] and explained well in [HTF09].

The steady-state base calibration is a very early step within the engine calibration process. Figure 1.1 illustrates the placement of steady-state ECU calibration phases within the overall engine development process. Further calibration tasks, e.g. the on-board diagnostics or the drivability calibration, take place within the development process. However, these are not addressed in this thesis. Especially the transient calibration process strongly differs from the steady-state one and will not be focused. The main difference regarding the measurement procedure is the necessity to excite the engine dynamically for transient calibration, which is not the case for a steady-state calibration. The steady-state calibration can be done in very different ways. Most of the methods to achieve a high quality have steady-state measurements in common but differ very much in the effort to be spent. Due to the curse of dimensionality, some methods are not applicable to tests in which a high amount of actuators are varied. A full factorial test design reaches its limits quickly, because a large amount of tests increasing exponentially with the dimension is necessary. A one-factor-at-a-time optimization will not find the optimal solution with high certainty, as interacting influences between the inputs are not taken into consideration. Thus the identification of a surrogate model is a widely used method to find the optimal settings offline by means of an engine simulation model with only little measurement effort. The ECU maps and curves then are filled in a way that the functions generate the optimal setting found offline.

A crucial step in this model-based calibration process is the test design composition, that defines the steady-state measurements necessary to train a statistical model. This test design could be static, identified in advance of all measurements, or dynamically updated by an adaptive strategy during

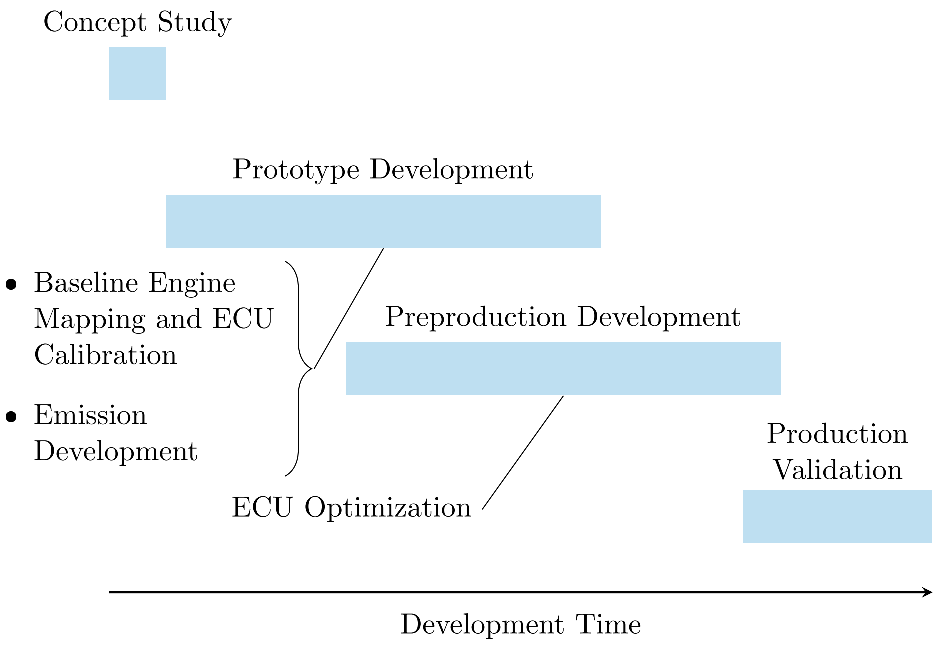


Figure 1.1: Schematic extraction from the engine development process as described in [HD16]

the measurement campaign. The benefit of an adaptive test design is clearly the possibility to involve system knowledge into the further test design and therefore improve the surrogate model quality outcome. However, there are two different ways to implement the whole process of determining an optimal solution for the actuator settings by a surrogate model, for which reason the objective of the adaptive design can be very different. These two approaches are the online optimization procedure, which implements the full process of optimal actuator setting search, and the engine model identification procedure, that aims at reproducing the combustion engine in a wide range as precise as possible for a subsequent offline actuator setting search.

1.1.1 Online Optimization

Finding an optimal actuator setting at a minimal possible calibration engineer intervention is the main objective of a calibration method development.

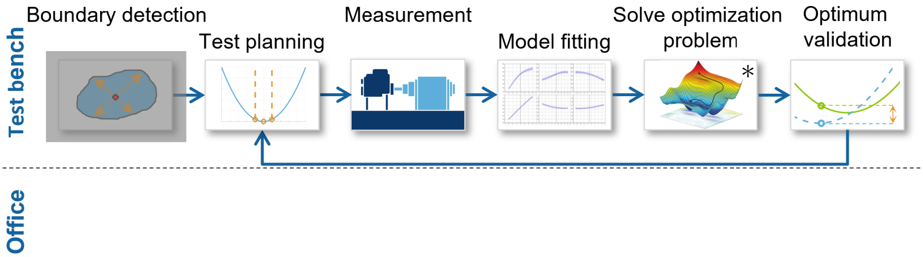


Figure 1.2: Schematic representation of an online optimization process. *Illustration source: [Ami+19]

An appropriate practice is to shift most of the work done offline in office to be executed by an algorithm online at the test bench during the engine tests. Hence, the process as shown in figure 1.2 arose that, in theory, only makes it necessary to define the system inputs, outputs and an optimization definition. In case unknown limitations are present, a boundary search is an optional preceding step before entering an iterative optimization loop. Within this loop, a test design stage defines the next measurements to be performed, followed by the execution of this test plan. A surrogate model is updated that is used to solve the optimization problem. The optimal setting validation step checks the quality of the currently identified optimal solution. In case the deviation between modeled and measured optimum is higher than allowed, a further refinement is necessary and a next iteration takes place.

This optimization process has its clear strength in automating most of the calibration process and additionally reduces the necessary measurements to find an optimal solution significantly. However, this process also has some drawbacks to be revealed. The most unacceptable property is the necessary definition of the optimization problem in advance and the focused measurement towards the identification of this optimal solution. In practice, several optimization loops take place during a steady-state actuator calibration. The criteria regarding the emission calibration can change due to an altered exhaust aftertreatment system for example. Also the running smoothness criteria could change once the engine is tested within the vehicle. Another human factor could and does often apply, which is an incomplete or erroneous optimization problem. The measurements conducted to solve the optimization problem strongly focus the problem itself.

A change of the optimization target thus forces an additional almost complete iteration including measurements at the test bench. To avoid any additional measurements once any target changes, the process could be split into model identification and optimization problem solving, as will be discussed in the following section.

1.1.2 Combustion Engine Model Identification

A necessary condition to solve an optimization problem with high quality is the presence of a most exact surrogate model possible of the combustion engine. The identification process of this model is clearly different and follows other targets than the online optimization process. Figure 1.3 illustrates the adaptive test design procedure for the data-driven model identification. An optimal solution is to focus an iterative test planning method that aims at identifying a model with high quality in the largest possible measurement range. Therefore the identification of engine boundaries, which could be done in advance of the model identification process, should be integrated into the iterative procedure as a test design criterion. The adaptive test design process then follows the steps test planning, which objective is to find test points that maximally contribute to the design targets, steady-state measurement of the previously defined test points, a model training or update at the test bench as well as a model validation to rate the current model quality and define a potential abortion criterion. As shown in figure 1.3 the optimization step is completely detached from the model identification process and is done subsequently in office.

In contrast to a non-adaptive approach with a boundary search in advance, the adaptive test planning target is a multicriterial optimization problem. The major aim is to find a design algorithm that fulfills the optimization targets best and also complies with the real-time constraint to not interrupt the measurement process at the test bench, which is the main focus of this thesis.

1.2 Objectives and Outline

The identification of a combustion engine model is an often described method in literature. Plenty of different non-calibration and calibration focusing test design, modeling and optimization algorithms have been in-

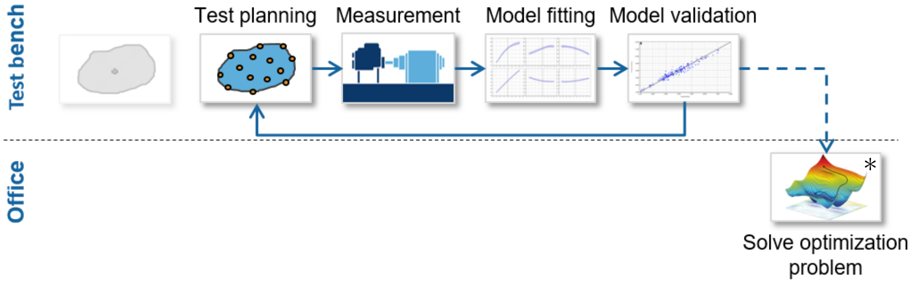


Figure 1.3: Schematic representation of the adaptive surrogate model identification process. The optimization problem solving process step, which aims at finding an optimal solution for the ECU calibration, is shown for the sake of completeness, even though it is not part of the encapsulated engine model identification process. *Illustration source: [Ami+19]

troduced that are applied in the field of combustion engine model identification. Boundary search is a described method as well with some applications focusing engine calibration. Adaptive test design methods are present in literature as well. However, combining the two given criteria boundary search and model quality focusing test design is a field seldom examined. For this reason, the existing methods will be analyzed, whereupon alternative and new approaches to improve the weaknesses will be discussed, implemented and tested. The scope of application is the base ECU calibration including emission development. Since the model-based calibration (MBC) applies within both, diesel and gasoline engine calibration, the methods have to be applicable within both processes. In addition to MBC topics, there are further applications for identifying an overall engine model within the engine development process. Hardware comparison studies in the advance development field shall be introduced as an example. Since a change in hardware, e.g. a different injector or an added tumble valve, could change the overall engine behavior, a comparison of several hardware settings always requires a full engine model including the influence of all other significant parameters. For instance, given an optimization criterion the optimal injection pattern could differ if the fuel injectors are changed. Hence, an interaction between injector type and injection pattern is present that cannot be neglected.

Next to the performance of algorithms, some more criteria apply in the scope of a process that a development staff has to apply. Developing hard-to-tune algorithms delivering excellent results applied by an algorithm expert should not be the focus within engine calibration. Calibration engineers commonly focus the thermodynamical processes of a combustion engine rather than the mathematics behind an applied algorithm. Additionally, time pressure within calibration projects prevents engineers to learn new complex methods [IS20]. Therefore, a model-based calibration process should fulfill the requirement of an easy application without the necessity to adjust non-tangible parameters.

This thesis is structured as follows. The subsequent chapter 2 examines the state of the art regarding statistical modeling, test design strategies, boundary detection methods, and currently existing adaptive test design methods. The exact research topics as well as detailed requirements for the development of a newly adaptive test design strategy will be described as well. In chapter 3, two open research topics are addressed that consider engine boundaries within an adaptive test design method. The developed methods will be discussed as well as tested in a simulation environment. Two further open questions in the field of a model-based adaptive test design strategy are addressed in chapter 4 without considering engine boundaries. Present and newly developed methods are compared in a simulation environment to find the most effective model-based test design approach. A combination of model-based design with considering engine boundaries in a user-friendly environment, based on the results of the preceding chapters, is developed in chapter 5. Its performance is again examined in a simulation environment. A real world test of the multicriterial approach at an engine test bench and its results will be shown in chapter 6. The final chapter 7 concludes the obtained results and gives a proposal for further research topics within the addressed field.

2 State of the Art

To consider and discuss the main topics of this study, the fundamental algorithms have to be introduced. The most important base of an adaptive test design method is the underlying model, which has to be trained by measurements. Different usual model classes, their advantages and disadvantages as well as the decision making are discussed in section 2.1. For each model type, different adjusted test design methods exist, to gain most knowledge and achieve a high-quality model, once trained. A review of different test design methods and their matching model types is given in section 2.2.

In contrast to computer experiments, real measurements are mostly constrained in the output domain. Since the output constraints restrict the input domain, it is beneficial to take boundaries into account during test design. The often non-known restrictions need to be identified and described. A detailed introduction of boundaries and the basics about the identification and description of those are captured in section 2.3.

The fourth section 2.4 deals with an overview of existing adaptive test design approaches. A conclusion is given regarding their general focus, utilized model classes, and test design methods. Since not all questions are answered by the existing procedures and in relation to the most beneficial model classes and test design methods, the open questions are discussed in the last section 2.5 and the topics of this research are outlined.

2.1 Data-Driven Combustion Engine Modeling

A combustion engine can be modeled by two main different types of models. A physically-based model can be established to represent the engine's behavior. This model type has the advantage that, once adjusted, it extrapolates well and an adjustment to changed engine hardware can be applied without rebuilding the whole model. However, the effort to compose a physically-based model is high and several necessary outputs cannot be explained well. Data-driven models are therefore utilized, because their ability to reproduce the combustion process and the exhaust emissions is much better. Given a model architecture matching the area of application, no deep model setup knowledge is necessary to train such a model, unlike physically-based models. These properties and the accurateness of data-driven models are the main reasons for their broad application in the field of combustion engine calibration and why they are investigated in this thesis. Several different data-driven model types are utilized in terms of combustion engine modeling. The most important and common model types are introduced in the following sections.

2.1.1 Polynomial Model

The polynomial model is a widely used approximation method for combustion engine processes. The model structure complexity is very low, which is why the physical interpretability by an engineer is given. Having a look at the relation between a measurement y and its representation by a polynomial model given by [Fah+13]

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \epsilon \tag{2.1}$$

with error term ϵ, the impact of each of the k inputs x_i is addressed by its belonging factor β_i. This representation assumes a linear behavior between the process output y and the inputs x_i, which will result in a high model error in engine modeling. However, the parameters x_i also can be seen as a substitution of input combinations of higher order. A polynomial model of degree l can be written as

$$y = \beta_0 + \sum_{i_1=1}^{k} \beta_{i_1} x_{i_1} + \sum_{i_1=1}^{k} \sum_{i_2=1}^{k} \beta_{i_1,i_2} x_{i_1} x_{i_2} + .. + \sum_{i_1=1}^{k} .. \sum_{i_l=1}^{k} \beta_{i_1,..,i_l} x_{i_1} .. x_{i_l} + \epsilon \tag{2.2}$$

which still is linear in the parameters β_i. If the product of the x_i is substituted by parameter t, (2.2) is expressed by

$$y = \sum_{i=0}^{m} \beta_i t_i + \epsilon \qquad \text{with} \quad t_0 = 1 \tag{2.3}$$

given the number of terms m. The calculation for the amount of parameters is given in [Nel20].

To train a fixed polynomial structure given some measurement data, the weights $\boldsymbol{\beta}$ need to be determined. Since the identification of a true parameter vector $\boldsymbol{\beta}$ is practically impossible due to a measurement and a bias error, as will be described later, the hat notation is introduced to declare the estimated weights $\hat{\boldsymbol{\beta}}$. Given n measurements, a regression matrix $\boldsymbol{M}$ and an estimated weight vector $\hat{\boldsymbol{\beta}}$ are represented by

$$\boldsymbol{M} = \begin{bmatrix} 1 & t_{1,1} & t_{1,2} & \cdots & t_{1,m} \\ 1 & t_{2,1} & t_{2,2} & \cdots & t_{2,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_{n,1} & t_{n,2} & \cdots & t_{n,m} \end{bmatrix} \qquad \hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta_0} \\ \hat{\beta_1} \\ \hat{\beta_2} \\ \vdots \\ \hat{\beta_m} \end{bmatrix} \tag{2.4}$$

The expected output then is expressed as a matrix multiplication by

$$\hat{\boldsymbol{y}} = \boldsymbol{M}\hat{\boldsymbol{\beta}}. \tag{2.5}$$

Due to the measurement and the bias error, a measurement $\boldsymbol{y}$ deviates from an estimation $\hat{\boldsymbol{y}}$, which is why a residual vector $\boldsymbol{e}$ is added to the modeled output

$$\boldsymbol{y} = \hat{\boldsymbol{y}} + \boldsymbol{e} = \boldsymbol{M}\hat{\boldsymbol{\beta}} + \boldsymbol{e} \tag{2.6}$$

declaring the residuals as the deviation between modeled output and measurement. The aim of a training is therefore to minimize the residual term. Since this model type is linear in its parameters, the least squares method can be applied. (2.6) is squared and rearranged to resolve for the squared residual vector

$$\boldsymbol{e}^\top \boldsymbol{e} = (\boldsymbol{y} - \boldsymbol{M}\hat{\boldsymbol{\beta}})^\top (\boldsymbol{y} - \boldsymbol{M}\hat{\boldsymbol{\beta}}). \tag{2.7}$$

A minimum error is determined by setting the first derivative of (2.7) to zero. The result is resolved for $\hat{\boldsymbol{\beta}}$

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{M}^\top \boldsymbol{M})^{-1} \boldsymbol{M}^\top \boldsymbol{y} \tag{2.8}$$

and can be solved simply. It can be shown, that a maximum likelihood (ML) estimation of the parameter vector $\boldsymbol{\beta}$ leads to the same result as the least squares method. Assuming the noise term normally distributed as

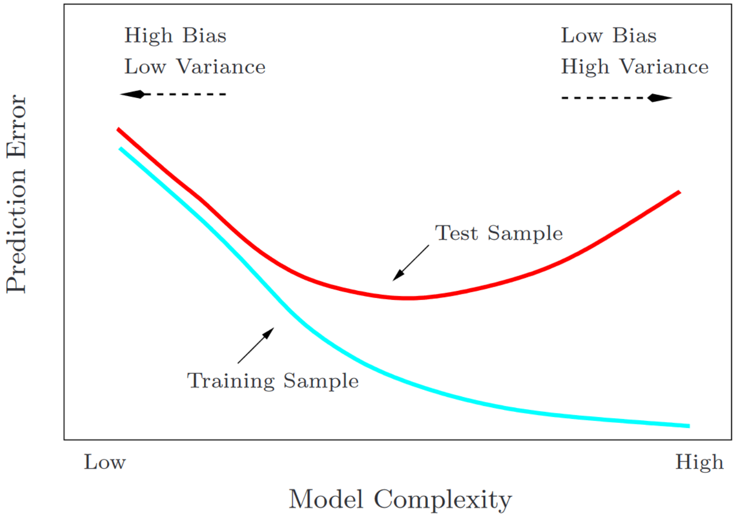


Figure 2.1: Bias Variance Trade-off [HTF09]

$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$, an ML estimation is applicable and results in a minimization of the least squares term (2.7) [Fah+13].

In many practical use cases a dense parameter vector leads to overfitting. The number of terms rises exponentially with the dimension and the polynomial degree. The flexibility of a model grows the same way, which is why it is increasingly able to explain measurement noise as process behavior. A model with low training error could be strongly overfitted and leads to a very high error in test data that is not used for training. Figure 2.1 shows this correlation. Very complex models show overfitting, which leads to a high variance in test data. If the complexity is low, both the training error and the test error are high. However, to apply polynomial models to high dimensional processes with suitable quality, term selection procedures are utilized. The methods described in literature [Fah+13; HTF09; Nel20; Jam+13] differ in their resulting quality and computational complexity. Also for combustion engine modeling, there is no appropriate method that applies best to all system outputs. Although polynomial models trained with a term selection procedure can reduce overfitting, the necessary flexibility in terms of nonlinearity to explain engine behavior in the whole speed and load domain is not given. Hence, the term selection procedures are not

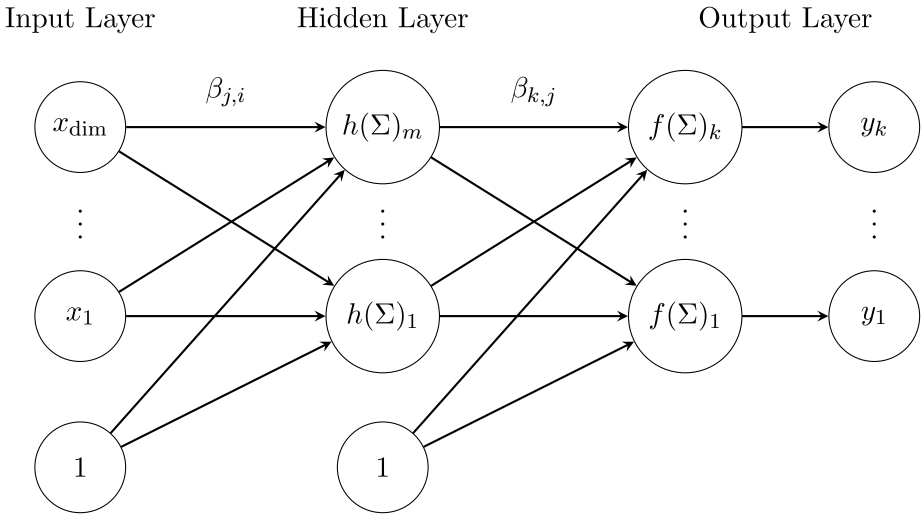


Figure 2.2: Single-hidden-layer or 3-layer artificial neural network architecture

described in detail but model types with higher flexibility are required. These are explained in the subsequent sections.

2.1.2 Artificial Neural Network

The artificial neural network is a model class that is based on the brain architecture of intelligent creatures. The idea from a signal processing point of view is to map a high amount of multi-dimensional input data to one or several outputs in the same way as a brain is able to. A schematic representation of a single-hidden-layer artificial neural network, also called 3-layer network, is given in figure 2.2. From a mathematical point of view, the input data $\boldsymbol{x}$ is mapped by an activation function $f(\cdot)$ and the inner nonlinear basis functions $\Phi_j(\boldsymbol{x})$ given a weight vector $\boldsymbol{\beta}$ to the output

$$y(\boldsymbol{x}, \boldsymbol{\beta}) = f\left(\sum_{j=1}^{m} \beta_j \Phi_j(\boldsymbol{x}) + \beta_0\right) \tag{2.9}$$

where m is the number of neurons in the hidden layer. In multilayer perceptron (MLP) networks another activation function $h(\cdot)$ is applied to the basis functions

$$y(\boldsymbol{x}, \boldsymbol{\beta}) = f\left(\sum_{j=1}^{m} \beta_j \, h\left(\sum_{i=1}^{\dim} \beta_{j,i} x_i + \beta_{j,0}\right) + \beta_0\right) \tag{2.10}$$

where dim is the number of input dimensions. If the output layer consists of more than one neuron, (2.10) extends to

$$y_k(\boldsymbol{x}, \boldsymbol{\beta}) = f\left(\sum_{j=1}^{m} \beta_{k,j} \, h\left(\sum_{i=1}^{\dim} \beta_{j,i} x_i + \beta_{j,0}\right) + \beta_{k,0}\right) \tag{2.11}$$

with k indexing the output number and system outputs respectively. A constant input neuron $x_0 = 1$ and a hidden neuron without an input can be applied, which simplifies the mapping to

$$y_k(\boldsymbol{x}, \boldsymbol{\beta}) = f\left(\sum_{j=0}^{m} \beta_{k,j} \, h\left(\sum_{i=0}^{\dim} \beta_{j,i} x_i\right)\right) \tag{2.12}$$

which represents the full MLP network architecture as shown in figure 2.2. In terms of model-based calibration, a regression is performed. Hence, the activation function of the output layer, which is necessary in classification application, is not used and a regression MLP network is defined by

$$y_k(\boldsymbol{x}, \boldsymbol{\beta}) = \sum_{j=0}^{m} \beta_{k,j} \, h\left(\sum_{i=0}^{\dim} \beta_{j,i} x_i\right). \tag{2.13}$$

The most common activation functions $h(\cdot)$ are the *logistic sigmoid* or the *tanh* function [Bis06]. The MLP is a very flexible network in case the number of hidden neurons m is high. Given the data, m needs to be defined prior to the training. In case a high amount of hidden neurons is given, the MLP tends to overfit the data and has a low prediction quality. In practical applications, either the amount of hidden neurons is parameterized manually or a high quantity is defined and a regularization term called weight decay is added to the error function during training. The regularization strength is defined by the value λ in the error function

$$\tilde{E}(\boldsymbol{\beta}) = E(\boldsymbol{\beta}) + \frac{\lambda}{2}\boldsymbol{\beta}^\top\boldsymbol{\beta} \tag{2.14}$$

and rewards small weights during training. The training of an MLP always is a nonlinear optimization problem. The activation function maps the input in combination with the weights in the hidden layer to a nonlinear output behavior. Therefore, the training of an MLP is very time consuming and often results in a local optimal solution if a local optimization procedure is applied [Nel06].

Next to the MLP network, radial basis function (RBF) networks [HJ01; Bis06] are often utilized. This type of neural network differs from MLP by the type of basis function $\Phi_j(\boldsymbol{x})$. While MLP creates nonlinearity by applying an activation function to the linear mapping, the RBF base is a symmetric basis function, which is activated by the distance to the center of each neuron. Introducing the radial basis function to (2.13) gives

$$y_k(\boldsymbol{x}, \boldsymbol{\beta}) = \sum_{j=0}^{m} \beta_{k,j}\, \Phi_j\left(||\boldsymbol{x} - \boldsymbol{c}_j||\right) \tag{2.15}$$

with a centered neuron at each location $\boldsymbol{c}_j$ and the vector norm $||\cdot||$. To avoid overfitting, typically fewer centers are defined than training points. This leads to more complex training because next to the weights and basis function parameters, the center locations need to be found. The most common radial basis function for the RBF network is the Gaussian function

$$\Phi_j(\boldsymbol{x}) = \exp(-\frac{||\boldsymbol{x} - \boldsymbol{c}_j||^2}{2\sigma_j^2}) \tag{2.16}$$

with the impact range of neuron j defined by the variance σ_j^2. The shape of each basis function is elliptic if a covariance matrix $\boldsymbol{\Sigma}_j$ is utilized instead of a fixed variance term σ_j^2

$$\Phi_j(\boldsymbol{x}) = \exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{c}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x} - \boldsymbol{c}_j)) \tag{2.17}$$

which significantly increases the amount of parameters to train. A compromise on amount of parameters and quality is a diagonal covariance matrix, where a rotation of the ellipsoids is prohibited but a dimension specific width still is present [Nel06].

Next to Gaussian shaped basis functions, thin-plate splines are sometimes applied [HJ01]. The basis function choice mainly depends on the process to approximate. A general judgment of a best-performing basis function is not applicable. Since the combustion engine outputs behave very different, even in terms of model-based calibration none of the basis functions applies best for all outputs.

The training of an RBF network strongly depends on the assumption of the basis functions definition. In case the location $\boldsymbol{c}_j$ and width $\boldsymbol{\Sigma}_j$ is optimized together with the weights $\beta_{k,j}$ a nonlinear optimization is present. In practical applications, different procedures are described, which first define the centers and then optimize the weights. The main advantage of the sequential optimization is the determination of the weights, which are calculated by least squares if the basis function is defined. The location of the centers is the most challenging problem, which is done for example by randomized selection, clustering, or a stepwise selection [Nel06].

2.1.3 Local Linear Model Network

A main advantage of RBF or MLP neural networks is their flexibility, which comes with a main disadvantage, namely the degree of freedom regarding parameterization and training. Driven by the target to design a flexible model class with low training and adjustment effort, local linear model networks are developed that are able to reproduce the combustion engine behavior within the whole engine operating range. The influence of actuators to outputs often changes strongly within the input domain, as for example injection parameters have varying influence on soot emissions within different operating areas and for a different fuel pressure. Polynomial models as described in section 2.1.1 are not able to represent this behavior accurately. Local linear model networks are introduced to divide the input domain into areas of high nonlinearities and model each zone with an own model of low complexity. The composition of those low complexity models is able to model more complex behavior. The base model structure is a neural network at which each zone is defined by a neuron and its activation function. Several different partitioning algorithms exist, as described in [Har14] for example. The two most important model network types in the segment of combustion engine modeling, which are based on tree structures, shall be presented.

Local Linear Model Tree

The local linear model tree (LOLIMOT) was introduced in [NSI96]. The idea behind this model structure is to separate the input domain axis-orthogonally in areas of strong nonlinearity. For each zone, a local polynomial model is trained and combined with the neighbor zones by a validity function. The base of the validity function is given by Gaussian functions in the input domain [Har14] as

$$\tau_i(\boldsymbol{x}) = \exp\left(\sqrt{(\boldsymbol{x} - \boldsymbol{c}_i)^\top \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{c}_i)}\right) \tag{2.18}$$

with the center of each function defined by $\boldsymbol{c}_i$ and the corresponding covariance matrix $\boldsymbol{\Sigma}_i$. Due to the axis-orthogonal separation, the covariance matrix for this model type is a diagonal matrix

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_{i,1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{i,2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{i,\mathrm{dim}}^2 \end{bmatrix} \tag{2.19}$$

given the variance in dim dimensions for the particular partial model i. A scaling of the Gaussian functions results in validity functions. The sum of all validity functions τ_i is always one, which is the so-called partition of unity:

$$\zeta_i(\boldsymbol{x}) = \frac{\tau_i(\boldsymbol{x})}{\sum_{j=1}^{M} \tau_j(\boldsymbol{x})} \tag{2.20}$$

An example of three Gaussian functions and the corresponding validity function is shown in figure 2.3. The training of a LOLIMOT consists of two parts. The polynomial model in each zone is trained by least squares with low effort. The identification of the zone locations takes more effort because a combined training of all splits results in a nonlinear optimization problem. That is why a tree structure is applied for LOLIMOT, which can be trained iteratively. The zone with the highest error is split into two zones in an axis-orthogonal way through the center. To find the direction all

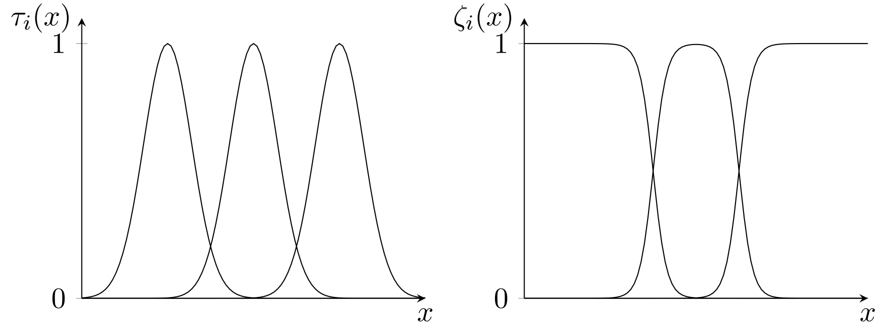


Figure 2.3: Gaussian membership function (left) and validity function (right) [Har14]

possible mutations are executed and the most improving one is used. The input domain is split until a defined abortion criterion is reached, which could be the maximum amount of zones, a defined training error, or a validation error. Figure 2.4 exemplarily shows the iterative split procedure in a two dimensional input domain. The width of each zone and also the smoothness of local model transition is defined by the covariance matrix Σ_i. For LOLIMOT, the $\sigma_{i,j}$ are defined by the dimension specific width of the corresponding local model Δ_{ij} and a smoothness factor of $k_\sigma = 1/3$ [Har14] to

$$\sigma_{i,j} = \frac{1}{3}\Delta_{ij}. \tag{2.21}$$

The mean advantages of the axis-orthogonal separation of the input domain are the good interpretability and the low amount of parameters to train. A projection of the Gaussian functions to one-dimensional representations enables practitioners to understand the partitioning in practical applications. However, this model type also has disadvantages in some cases. Given a process with strong nonlinear behavior in the direction of the input axes the splitting works very well and the process behavior can be explained with low effort and high quality. In case high changing gradients exist on the input domain diagonal, many axis-orthogonal separations are necessary to explain the behavior [Har14]. This is not only a problem of training duration but mainly a problem of necessary amount of measurement. The more splits that are performed, the more training data is necessary to provide enough data for each local model. For this reason, the hierarchical local model tree is introduced and presented in the following.

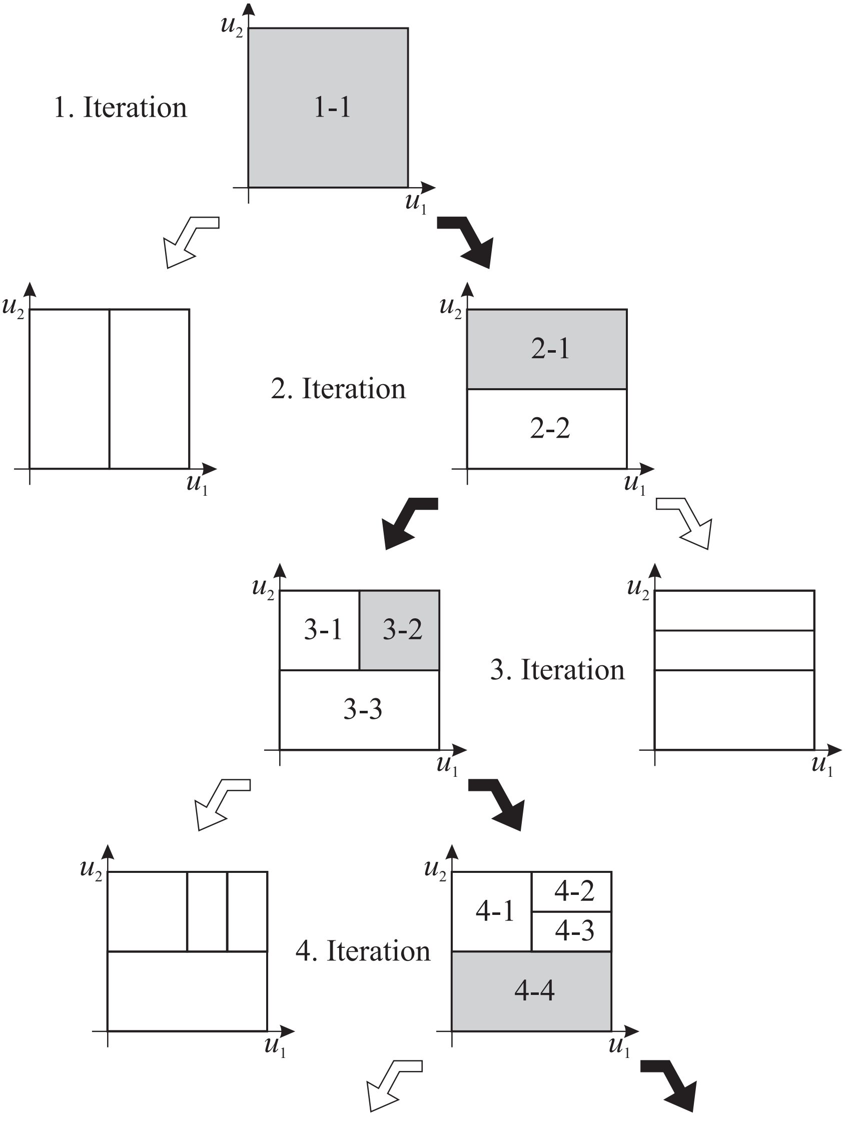


Figure 2.4: Exemplary LOLIMOT partitioning sequence with 4 iterations in a two-dimensional input domain with inputs u_1 and u_2 [Nel20]

Hierarchical Local Model Tree

The main difference between the described LOLIMOT algorithm and the hierarchical local model tree (HILOMOT) is the way the separation is done in the input domain. While LOLIMOT only searches for axis-orthogonal, centric splits, the splitting hyperplane can be located axis-oblique anywhere in the considered zone with the HILOMOT. Unlike LOLIMOT, the search for an axis-oblique partitioning is a nonlinear optimization problem. A dense, symmetrical covariance matrix is the result if Gaussian functions are used as base for the validity function. Following multilayer perceptron networks, usually sigmoid functions

$$\zeta_i(\boldsymbol{x}) = \frac{1}{1+\exp(\alpha_i)} \tag{2.22}$$

with the exponent

$$\alpha_i = -\kappa \left[1 \ \boldsymbol{x}\right]^\top \boldsymbol{p}_i \tag{2.23}$$

are used to identify the validity function for each split. α_i defines the smoothness of the sigmoid function and mainly controls the transition between the zones. Since this function type is fully defined by the split direction parameter vector $\boldsymbol{p}_i$ with $n+1$ entries for each split and the smoothness constant κ, fewer parameters need to be optimized as compared to Gaussian functions [Har14].

The training is also iteration-based and starts with a one-zonal polynomial model. In each iteration, the worst zone is identified by model error and a split is executed. The initialization subset of hyperplanes for the split is generated by all axis-orthogonal hyperplanes and one in the same direction of the preceding split hyperplane, passing through the center of the zone. Starting with the hyperplane from the initialization subset that leads to lowest model error, the split direction and location is optimized in a nonlinear way. As optimization criterion the normed root mean squared error (NRMSE) of the training data is used. The same abortion criteria as for LOLIMOT apply for the HILOMOT training as well. Figure 2.5 shows an exemplary axis-oblique splitting procedure of the input domain by using the HILOMOT algorithm.

Especially for high dimensional process behavior, as it occurs in combustion engine calibration applications, HILOMOT shows better results regarding the model error compared to LOLIMOT [Har14]. The axis-oblique

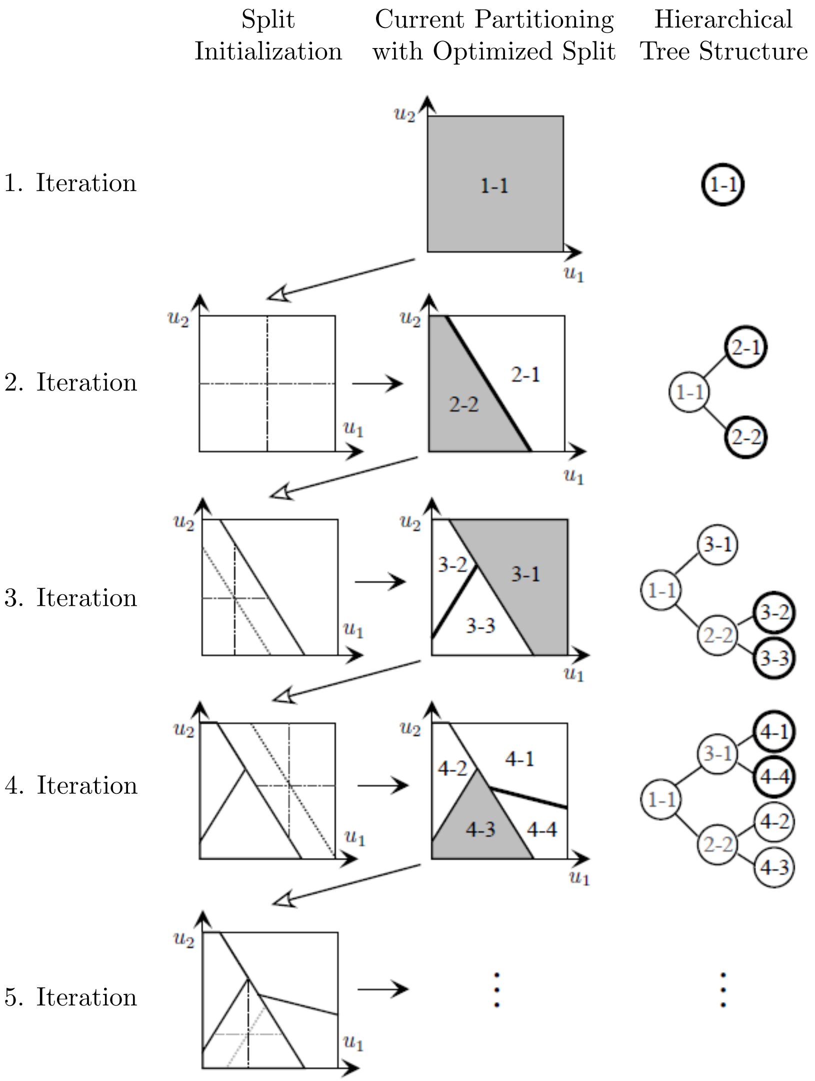


Figure 2.5: Exemplary HILOMOT partitioning sequence with 5 iterations in a two-dimensional input domain with inputs u_1 and u_2 [Har14]

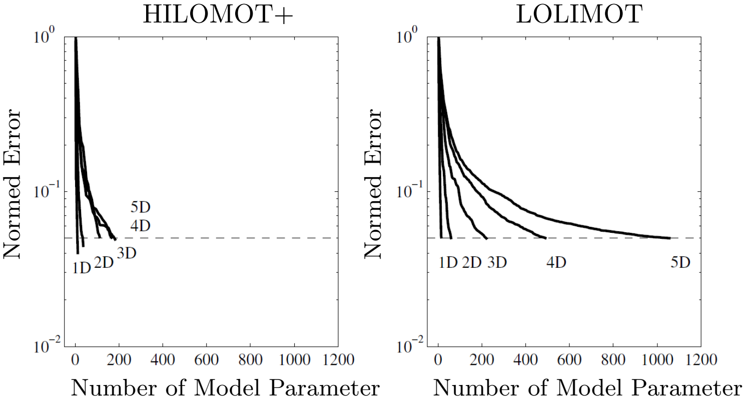


Figure 2.6: Comparison of model error over number of model parameters, which is the sum of partitioning and local model parameters, for HILOMOT+ (left) and LOLIMOT (right) for input dimensions one to five [Har14]

split direction highly improves the model flexibility, which is necessary for nonlinear process modeling. A further improvement is achieved by an additional term selection procedure during the HILOMOT training as also introduced in [Har14]. During each iteration, the significant terms of the considered local polynomial model of given maximum order are selected and the necessity of a split is considered. The improvement of this so-called HILOMOT+ compared to the classical LOLIMOT model is shown in figure 2.6. The amount of model parameters at the same error level is significantly reduced, which increases exponentially with the dimension.

2.1.4 Gaussian Process Model

The utilization of the Gaussian process model (GPM) in model-based engine calibration has a young history. Even though it was introduced in 1960 under the name Kriging in the area of meteorology and geostatistics as an interpolation method for spatial modeling, the GPM in terms of machine learning was introduced in 1996 by Williams and Rasmussen [WR96]. The popularity was achieved by the publication of Rasmussen and Williams in 2006 [RW06] where the basics, algorithmic improvements, and also an implementation is described. This made the complex probabilistic theory accessible for a broader engineering application range. Also the implementation details significantly improved the training performance, which is crucial for the expensive training of a GPM. The first important comparison of GPM and state-of-the-art-utilized model classes in engine calibration was made in 2011 by Berger, Rauscher and Lohmann [BRL11] and Berger and Rauscher [BR11] respectively. Earlier publications indicate the introduction of GP-based modeling techniques, as for example Kruse, Ulmer and Schulmeister in 2007 [KUS07], but do not mention the GPM directly.

The derivation of a GPM and its similarity to other model types is most easy to understand if the Bayesian linear model is introduced first. The base of this model is again the linear model

$$f(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{\beta}. \tag{2.24}$$

If the inputs first are projected into feature space, the linear model is able to represent nonlinearities, but the model stays linear in its parameters and takes the form

$$f(\boldsymbol{x}) = \Phi(\boldsymbol{x})\boldsymbol{\beta} \tag{2.25}$$

with the basis function $\Phi(\boldsymbol{x})$. Noise is added to the function

$$y = f(\boldsymbol{x}) + \epsilon \tag{2.26}$$

as independent, identically distributed Gaussian noise with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, given the noise variance σ_n^2. The noise assumption is important for engineering applications, as the function values will always differ from experiment results. In a Bayesian framework, inference on the parameters is made by Bayes' rule

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}. \tag{2.27}$$

Hence, the likelihood and the prior for the linear model needs to be derived. Given the Gaussian distributed noise, the likelihood is defined by

$$\begin{aligned} p(\boldsymbol{y}|\Phi(\boldsymbol{X}), \boldsymbol{\beta}) &= \prod_{i=1}^{n} p(y_i|\Phi(\boldsymbol{x}_i), \boldsymbol{\beta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp(-\frac{(y_i - \Phi(\boldsymbol{x}_i)\boldsymbol{\beta})^2}{2\sigma_n^2}) \\ &= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp(-\frac{1}{2\sigma_n^2}|\boldsymbol{y} - \Phi(\boldsymbol{X})\boldsymbol{\beta}|^2) \\ &= \mathcal{N}(\Phi(\boldsymbol{X})\boldsymbol{\beta}, \sigma_n^2\boldsymbol{I}) \end{aligned} \tag{2.28}$$

and gives rise about the probability of the measurements $\boldsymbol{y}$ given the parameters $\boldsymbol{\beta}$. The prior probability distribution needs to be conjugate to the likelihood, which in this case is a Gaussian distributed prior on the weights

$$\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_p) \tag{2.29}$$

with zero mean and covariance matrix $\boldsymbol{\Sigma}_p$. Only the likelihood and the prior depend on the weights $\boldsymbol{\beta}$. Therefore, the posterior probability is only dependent on the enumerator in (2.27) and one can write

$$p(\boldsymbol{\beta}|\Phi(\boldsymbol{X}), \boldsymbol{y}) \propto p(\boldsymbol{y}|\Phi(\boldsymbol{X}), \boldsymbol{\beta})p(\boldsymbol{\beta}). \tag{2.30}$$

Due to the Gaussian posterior and its conjugate Gaussian prior, the terms of the prior and likelihood that depend on the weights are necessary solely. The constant values are neglected and after applying some mathematics the resulting posterior is

$$p(\boldsymbol{\beta}|\Phi(\boldsymbol{X}), \boldsymbol{y}) \sim \mathcal{N}(\frac{1}{\sigma_n^2}\boldsymbol{A}^{-1}\Phi(\boldsymbol{X})\boldsymbol{y}, \boldsymbol{A}^{-1}) \tag{2.31}$$

with $\boldsymbol{A} = \sigma_n^{-2}\Phi(\boldsymbol{X})\Phi(\boldsymbol{X})^\top + \boldsymbol{\Sigma}_p^{-1}$. The complete derivation of (2.31) can be gathered from [RW06].

The most probable solution for the weights $\boldsymbol{\beta}$, also called the maximum a posteriori (MAP), is given by (2.31). A procedure to make predictions with this distribution is to calculate the mean value for the weights and use (2.25) to make predictions by

$$f(\boldsymbol{x}_*) = \Phi(\boldsymbol{x}_*)\bar{\boldsymbol{\beta}} \tag{2.32}$$

with $\bar{\boldsymbol{\beta}}$ as the mean of (2.31) and $\boldsymbol{x}_*$ as the input data to make predictions for. This gives rise about the mean value but one would miss the variance of the prediction. Therefore, in the Bayesian setting a prediction is made by putting a weight on each possible weight value by using the MAP distribution and average over the outcome

$$\begin{aligned} p(f_*|\Phi(\boldsymbol{x}_*), \Phi(\boldsymbol{X}), \boldsymbol{y}) &= \int p(f_*|\Phi(\boldsymbol{x}_*), \boldsymbol{\beta}) p(\boldsymbol{\beta}|\Phi(\boldsymbol{X}), \boldsymbol{y}) d\boldsymbol{\beta} \\ &= \mathcal{N}(\frac{1}{\sigma_n^2}\Phi(\boldsymbol{x}_*)^\top \boldsymbol{A}^{-1}\Phi(\boldsymbol{X})\boldsymbol{y}, \Phi(\boldsymbol{x}_*)^\top \boldsymbol{A}^{-1}\Phi(\boldsymbol{x}_*). \end{aligned} \tag{2.33}$$

The resulting distribution again is Gaussian with mean value same as entering the MAP mean to (2.32) but with given variance.

The Bayesian linear model is a good choice if the basis function architecture $\Phi(\boldsymbol{x})$ is given and the parameters need to be determined. It is robust concerning overfitting, because the prior distribution

$$p(\boldsymbol{\beta}) = \frac{1}{\sqrt{2\pi\boldsymbol{\Sigma}_p}} \exp(-\frac{1}{2}\boldsymbol{\beta}^\top \boldsymbol{\Sigma}_p \boldsymbol{\beta}) \tag{2.34}$$

acts as a penalty, reducing the posterior probability for high weights. This regression also is called ridge regression [HK70]. However, in many applications, and also in combustion engine modeling, the best matching basis function is not known in advance. At this point the GPM takes place. Instead of making inference in weight space, this model type directly infers the function in function space. Therefore, only a mean function $m(\boldsymbol{x})$ and a covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ is required and the Gaussian process is defined as

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')). \tag{2.35}$$

In many applications, the mean function is set to zero. The most common use to generate the covariance matrix is the squared exponential (SE) covariance function

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sigma_f^2 \exp(-\frac{1}{2l^2}|\boldsymbol{x_i} - \boldsymbol{x_j}|^2) + \sigma_n^2 \delta_{ij} \tag{2.36}$$

with the Kronecker delta δ_{ij} being one if and only if $i = j$ and zero otherwise. σ_n^2 represents the measurement noise variance and is added only to the training inputs, because again independent identically distributed Gaussian noise is assumed. The parameter l^2 is called length scale and controls the smoothness of the process. σ_f^2 defines the overall signal variance. As described, one important disadvantage of the Bayesian linear model is the basis function, which needs to be defined in advance. In [RW06], it is shown that the SE covariance function used in a Gaussian process behaves in the same way as a Bayesian linear model with an infinite number of basis functions.

Given the covariance function, the prior distribution over functions $p(f|\boldsymbol{X})$ is defined by $\mathcal{N}(\mathbf{0}, \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}))$. To make predictions only from the prior, the joint distribution of training outputs and test outputs

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I} & \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_*) \\ \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}) & \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) \end{bmatrix}\right) \tag{2.37}$$

is used. $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ defines the covariance matrix of all training inputs with each other. $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_*)$ is the covariance matrix of all training inputs and the test inputs and $\boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*)$ is the covariance matrix only of the test inputs. The identity matrix $\boldsymbol{I}$ has one entry on its main diagonal and zeros elsewhere. As this distribution is a joint Gaussian, the distribution between training and test inputs is Gaussian as well, which is an important characteristic.

In making predictions, the prior distribution is not of high importance. More interesting is the posterior distribution over functions, which incorporates the training outputs into the prediction. Therefore, the joint Gaussian distribution needs to be conditioned on the observations, which gives

$$\begin{aligned} f_*|\boldsymbol{X}, \boldsymbol{X}_*, \boldsymbol{y} \sim \mathcal{N}(&\boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}]^{-1} \boldsymbol{y}, \\ &\boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X}_*) - \boldsymbol{K}(\boldsymbol{X}_*, \boldsymbol{X})[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) \\ &+ \sigma_n^2 \boldsymbol{I}]^{-1} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_*)). \end{aligned} \tag{2.38}$$

Given this multivariate Gaussian distribution, the GPM is fully defined and predictions can be made given measurement data. In Gaussian process regression the main goal is not to find weights, because the function space view does not map weights to training data but infers directly from training data to prediction. A prediction in the Bayesian linear model as in (2.33) is only dependent on the weights and not on the training data anymore. This is different in the GPM, as the mean in (2.38) is still dependent on the training data but without any weights. However, to train a GPM it is necessary to tune the covariance function. In case of the SE covariance function (2.36), the length scale, measurement noise, and signal variance need to be optimized. These parameters are called hyperparameters and therefore the training differs from finding weights in a linear model.

Two optimization procedures are mainly utilized. As typical for Bayesian settings, the maximum likelihood procedure applies to GPM as well. An alternative approach is to optimize the predictive probability in a cross-validation setup. Within this research, the length scale parameter l^2 has one entry for each input dimension and is therefore a vector of length dim, which is the amount of inputs. The input specific value enables the GPM to offer a variable flexibility in each dimension, which is necessary for an application in model-based calibration.

Marginal Likelihood Optimization

In contrast to the linear parametric model, the GPM has no weight parameters. The only way of adjusting the model to the data are the hyperparameters of the SE covariance function (2.36). One may first think of a maximization of the hyperparameter posterior

$$p(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{X}) = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y}|\boldsymbol{X})} \tag{2.39}$$

but there is no appropriate solution to approximate $p(\boldsymbol{y}|\boldsymbol{X})$. Hence, the likelihood $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})$ is optimized, which in the application of a GPM is called marginal likelihood. This is due to the fact that this term is the marginal likelihood in the posterior over parameters following Bayes' rule

$$p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta})p(\boldsymbol{\beta}|\boldsymbol{\theta})}{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}. \tag{2.40}$$

The marginal likelihood of a Gaussian process is thus defined by the integral over the likelihood of the data times the prior over functions

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \int p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X})p(\boldsymbol{f}|\boldsymbol{X})d\boldsymbol{f} \tag{2.41}$$

and gives rise about the probability of the data given the inputs.

The prior does not depend on the hyperparameters directly but is affected by them through the covariance matrix of the training data $\boldsymbol{K} := \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ in its Gaussian distribution

$$p(\boldsymbol{f}|\boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}) \tag{2.42}$$

and hence the marginal likelihood also depends on the hyperparameters. A maximization of the marginal likelihood is possible by changing the hyperparameter vector $\boldsymbol{\theta}$. The likelihood of the data is given by a factorized Gaussian

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{f}, \sigma_n^2 \boldsymbol{I}) \tag{2.43}$$

leading to the log marginal likelihood

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{y} - \frac{1}{2} \log |\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}| - \frac{n}{2} \log 2\pi. \tag{2.44}$$

The second term $1/2 \log |\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}|$ acts as a regularization and penalizes complex functions. Even though the marginal likelihood optimization is penalized, it still is an optimization only on the training data and takes all training data into account. This can lead to significant overfitting, which is why the cross-validation training is introduced.

Cross-Validation Optimization

The cross-validation method takes place in many model selection methods and generalization assessment of trained models [HTF09]. The most popular method is the leave one out cross-validation (LOOCV) procedure. To judge the generalization quality of a given model structure, it is trained $n-1$ times each with $n-1$ training data, leaving out each training sample once. The mean of all prediction errors of the omitted training sample is the LOOCV error. In terms of a model, the weights $\boldsymbol{\beta}$ of the model are determined in each training with fixed basis function. However, the procedure in Gaussian process regression is slightly different, because the model consists of hyperparameters and does not estimate weights from the training samples but uses the full data for predictions. In Gaussian process LOOCV error estimation, the hyperparameters stay fixed and only one training point is excluded from prediction respectively. The leave one out log predictive probability becomes

$$L_{LOO}(\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \log p(y_i|\boldsymbol{X}, \boldsymbol{y}_{-i}, \boldsymbol{\theta}) \tag{2.45}$$

with the predictive log probability for training sample i

$$\log p(y_i|\boldsymbol{X}, \boldsymbol{y}_{-i}, \boldsymbol{\theta}) = -\frac{1}{2}\log\sigma_i^2 - \frac{(y_i - \mu_i)^2}{2\sigma_i^2} - \frac{1}{2}\log 2\pi. \tag{2.46}$$

The mean μ_i and variance σ_i^2 are calculated by (2.38) where training sample i is removed from $\boldsymbol{y}$ and $\boldsymbol{X}$ but is used for $\boldsymbol{X}_*$ instead. In contrast to the marginal likelihood optimization, the leave one out log predictive probability maximization provides a solution to train the hyperparameters avoiding excessive overfitting. Since simplifications are present to solve (2.45), the computational effort is only slightly higher compared to (2.44). However, solving the derivative of the leave one out log predictive probability is more time consuming, leading to a higher time demand to train the hyperparameters compared to the marginal likelihood optimization [RW06].

2.1.5 Model Type Comparison

An important decision to make during this research is the choice of model type to develop the methods for. The test design is the main objective of an adaptive test design. In section 2.2, the criteria regarding training point distribution for the different model types will be introduced. Since these can be unequal, the model choice defines the algorithms that apply best during adaptive test design. Hence, a comparison of the introduced model types and the selection of the most suitable one form the base for this development.

The objective of model choice is not to select the best suitable type for an efficient adaptive test design but more to identify the best suitable one for model-based calibration. However, the model also should be applicable to an adaptive test design procedure. For instance, a model training without user interaction must be possible and therefore is treated as constraint for the selection. Some studies are already carried out regarding model comparison in terms of model-based calibration. The results will be shown and extended in this section.

The criteria for the selection mainly follow the studies of [BRL11] and [Tie15]. In the study of Berger [BRL11], the GPM is compared to polynomial models, MLP networks and the tree structure models LOLIMOT and HILOMOT. The common criteria to evaluate within model-based calibration are

- applicability to high-dimensional problems,
- amount of necessary training data,
- ability to represent strong nonlinear process behavior (accuracy),
- robustness against overfitting,
- approved for automatic model training,
- uncertainty representation.

A more detailed comparison is done by [Nel20]. This comparison is extended in [Tie15] by GPM in a dynamic system identification scope. Since the evaluation criteria are very similar for steady-state and dynamic modeling, the comparison can be transferred and adjusted. Additional to the 6 already shown criteria from [BRL11], further model-based relevant criteria, rated in [Tie15], are

- sensitivity to noise,
- interpretation,
- incorporation of prior knowledge,
- training speed / training of large data set,
- uneven data distribution,
- extrapolation behavior.

There exist many more criteria, which are neglected in the scope of adaptive test design. Criteria like iterative modeling and model recalibration seem to be relevant but as long as a new model training is possible they are not for an adaptive test design. The only disadvantage of bad model recalibration capability is the training speed, because a full training always needs more computational effort. The criterion sensitivity to noise is part of the model accuracy and therefore will be combined. Additionally to the criteria, weights are introduced. Most important are the model performance criteria accuracy and overfitting as well as the applicability to high-dimensional problems, as they frequently occur in model-based calibration. Measurements are expensive in MBC and the amount of necessary data is therefore a crucial criterion as well. A medium weight is introduced for criteria that are important but less significant than the performance topics. Automatic model training is relevant to enable non-experts to train a model with high quality and especially apply a model to an adaptive test design strategy. To achieve reliable results, the model uncertainty representation is a relevant criterion as well. A high training speed is important for a manual iterative optimization and for an adaptive test design strategy that trains a model several times during the measurement campaign. The model sensitivity to the sample distribution correlates with the amount of necessary data but can be influenced by a test design strategy and therefore obtains a low weight. Interpretability and prior knowledge incorporation are seldom demanded in MBC. Interaction plots are used to verify the model behavior and replace the model interpretation necessity. Prior knowledge can be incorporated for example during the test design by restricting the design area. Since data-driven models are not able to extrapolate accurately compared to physically-based models, extrapolation is mostly restricted in MBC applications by only incorporating the measured region during evaluation. Therefore, the extrapolation behavior is weighted low as well. Table 2.1 shows the rated model classes and the weights respectively.

The GPM performs very well in most criteria with a final sum of 45 and a weighted mean of 4.1, outperforming the other model types in both weighted and unweighted result. Especially the highly important topics accuracy, high-dimensional application, and necessary training data are very high rated for the GPM. The good accuracy can be explained by a comparison to MLP: The use of a GPM can be interpreted as an MLP with infinite amount of neurons or basis functions, respectively [Nea96; RW06]. Hence, the basic flexibility of a GPM does not depend on the architecture like with all other model classes. Even in HILOMOT, a local model type needs to be specified. However, the flexibility of a HILOMOT and also MLP with a high amount of neurons is sufficient for model-based calibration, which is why they are rated very well. Only polynomial models suffer from their low complexity. A raise of model order in polynomial models yet results in higher oscillation and therefore in overfitting and a wrong representation of the true process. Overfitting can be handled well by statistical term selection but the oscillation behavior would stay. In model-based calibration, the amount of measurements is one critical

Table 2.1: Rating of different model types. The higher the value, the better is the rating with 1 as worst and 5 as best rating.

	Weight	Polynomial	NN	HILOMOT	GPM
Accuracy	3	3[1]	4[1]	4[2]	5[2]
High-Dimensional Application	3	2[1]	4[1]	4[2]	5[2]
Amount of Necessary Data	3	3[3]	5	2[3]	5[3]
Overfitting	3	5[3]	3	4	3
Automatic Model Training	2	3	3	5	4
Uncertainty Representation	2	2[2]	1[2]	3[2]	5[2]
Training Speed	2	4[1]	1[1]	4[2]	2[2]
Training Sample Distribution	2	1[2]	5[2]	3	3[2]
Interpretation	1	3[1]	2[1]	4[2]	5[2]
Prior Knowledge Incorporation	1	2[1]	2[1]	4[2]	3[2]
Extrapolation	1	1[2]	3[2]	3[2]	5[2]
Sum	23	29	33	40	45
Weighted Mean	-	2.8	3.3	3.6	4.1

[1] ∼ [Nel20]   [2] ∼ [Tie15]   [3] ∼ [BRL11]

task to address. Measurements are done at a very cost-intensive engine test bench, which is why a high weight is put on the criterion *amount of necessary data*. Especially the HILOMOT model has a low rating, because the training is performed in each zone only with the belonging training data. Since the individual local model does not know about the process behavior in the neighboring zone, this property can lead to an inaccurate representation in the zone transition areas. GPM and neural networks in contrast take all measurement into account for model training. Due to the amount of coefficients, polynomial models need a high amount of data for complex process assumptions. A comparison of model quality at a low amount of training data is derived in [BRL11] where the GPM shows significantly better results as compared to a polynomial model. A contrary result is found in [Ran+13] where the polynomial model is rated slightly better regarding the necessary amount of training data compared to a GPM. However, the GPM still will be rated high, because the amount of necessary data rises with the complexity of the process and polynomial model complexity, respectively. The rating of polynomial models and GP models in [Ran+13] regarding accuracy is similar to [Tie15] and [Nel20].

In modeling with GPM and neural networks, overfitting is a significant issue to cope and needs to be taken into account strictly. Automatic model training is important in terms of an adaptive test planning procedure, because it relies on a trained model without user interaction. Polynomial models and neural networks in particular suffer from the fixed basis function. The split algorithm in HILOMOT is a nonlinear optimization and therefore only the polynomial order of the local models needs to be defined. Low order local models already achieve very flexible global model results [Har14]. Alternatively, the HILOMOT+ with automatic term selection could be applied. The only drawback of GPM regarding automatic model training is overfitting. Therefore, regularization needs to be considered accurately.

The training speed of a model becomes less and less important. Computer hardware is continuously getting more powerful and the main training operations are easy to be solved by parallel computing processes for all model classes. Regarding the adaptive test design the training speed needs to be taken into account, because the model needs to be trained for further test point calculations. The effort to train the GPM is high due to a necessary matrix inversion. However, this is not a strong restriction for the application but more a boundary constraint to be observed. More relevant is the training sample distribution. Neural networks are rated to be very robust for uneven distribution by [Tie15], but since all these model types are black box models it is hard to predict the true process in unseen areas, even in terms of interpolation.

Having a look at the overall rating, the Gaussian process model performs best and is closely followed by the HILOMOT. It is easy to judge which model type suits best for model-based calibration in most cases, because the main disadvantages of the GPM are training speed, training sample distribution, prior knowledge incorporation, and overfitting. As already stated, training speed is becoming less and less relevant. Training sample distribution must be addressed precisely by the test design and is a main target of the adaptive test design methodology anyway. Regularization could be considered during model training to cope with overfitting in case the LOOCV optimization does not fulfill the requirements. The incorporation of prior knowledge for test planning is a topic that will be discussed for GPM within this research. The most decisive features good accuracy and low amount of necessary training data are perfectly fulfilled by the GPM.

2.2 Model Specific Test Design Requirements

Section 2.1.5 deals with the comparison of different model types, which are typically applied in model-based calibration processes to describe the behavior of combustion engines. The training sample distribution is one aspect for the comparison of the model types as shown in table 2.1. Each model type reacts different to optimal distributed measurements where the optimum is again dependent on the model type. For that reason, the design of experiments (DoE) method is introduced with different types of test designs. The first methodology for designing an experiment was already introduced in 1926 by Ronald Fisher [Fis26]. Today, many different methods exist to plan experiments, suited to the model class that shall be applied to the measurements. Four different general model classes are introduced in section 2.1. The matching test planning methods are discussed in the following.

2.2.1 Optimal Test Design

The optimal test design is a group of different test point distribution creation methods, which all have different optimization objectives. The most popular and widely used one is the D-Optimal test design. As any optimal test design, it is only applicable to a given parametric model structure. The D-Optimal test design optimizes the test points to have a distribution that shows a minimized variance of model parameters after model training [Mon08]. All optimal designs are based on the Fisher information matrix $(\boldsymbol{M}^\top \boldsymbol{M})^{-1}$, which is part of the least squares estimate for model coefficients in linear modeling (refer to (2.8)). The D-Optimal test design can be calculated by minimization of the determinant of the Fisher information matrix or, more suitable, by maximization of the inverse

$$\boldsymbol{M}_{opt,D} = \underset{\boldsymbol{M}}{\operatorname{argmax}}(\det(\boldsymbol{M}^\top \boldsymbol{M})). \tag{2.47}$$

Another popular test plan creation method is the G-Optimal test design, which minimizes the maximum prediction variance. The test design is created by computing a solution for $\boldsymbol{M}$ that minimizes the maximum value of the hat matrix [Sch96; SBH10]

$$\boldsymbol{M}_{opt,G} = \underset{\boldsymbol{M}}{\operatorname{argmin}}(\max(\boldsymbol{M}(\boldsymbol{M}^\top \boldsymbol{M})^{-1}\boldsymbol{M}^\top)). \tag{2.48}$$

Several further optimal test design methods exist, such as A-, C- and E-Optimal designs to minimize the model coefficient variance and I- and V-Optimal designs to minimize the prediction variance [Kow17].

What all test design methods have in common is that there is no closed solution computable. The main procedure to generate an optimal test design is a candidate exchange method. A start design is initialized and then test points are exchanged iteratively from the design matrix by a candidate set. The most suitable solution will be declared as optimal solution, even if the global optimum is not found in most cases. An example for an efficient generation of D-Optimal test designs is the DETMAX-Algorithm, introduced by Mitchell [Mit74], which iteratively adds or substitutes points from the design matrix, leading to a higher determinant value of the information matrix in each iteration. This algorithm was adjusted by Welch to cope G- and V-Optimal designs as well [Wel84]. Several further algorithms with local and also global optimization procedures exist, which can be found in literature.

The optimal design procedure is often applied to linear parameter models with given basis functions. Therefore, they are mainly applied in polynomial model utilization. The more general formulation of the Fisher information matrix [Kay93] given a data set with N measurements and assuming Gaussian noise is

$$[\mathcal{I}(\boldsymbol{\beta})]_{ij} = \frac{1}{\sigma^2} \sum_{k=1}^{N} \frac{d\hat{y}_k}{d\boldsymbol{\beta}_i} \frac{d\hat{y}_k}{d\boldsymbol{\beta}_j} \tag{2.49}$$

and is applicable also to models that are nonlinear in their parameters. However, to find a test design becomes more expensive. Also the choice of model parameters e.g. in neural networks in practice is inappropriate. Given some rough prior knowledge, the split locations in LOLIMOT models can be defined [MTI03; Sta+11], whereupon an optimal design can be found prior the experiment.

2.2.2 Bayesian Experimental Design

The major problem in experimental design is the uncertainty about the model parameters that need to be determined. As Cochran [Coc73] pointed out very well in his article about experiments for nonlinear functions: "You tell me the value of θ and I promise to design the best experiment for estimating θ", which shows that experiments are easily designed if the parameters are known. θ in this sense are the model parameters, stated as $\boldsymbol{\beta}$ in this research. However, in most applications the aim of experiments is to gather information to determine the unknown parameters.

Most important in Bayesian experimental design is the choice of a utility function, which decides about the optimization criterion. One common method for linear model types is the Bayesian D-Optimal test design, which is strongly related to the non-Bayesian D-Optimal test design presented in section 2.2.1. The Bayesian D-Optimal test design originates from using Shannon information and therefore from the maximization of the Kullback-Leibler divergence between the prior and posterior distribution

$$U(\boldsymbol{X}) = \int \log(p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}))p(\boldsymbol{y}, \boldsymbol{\beta}|\boldsymbol{X})d\boldsymbol{\beta}d\boldsymbol{y} \tag{2.50}$$

as base for the utility function [CV95]. This leads to the design criterion

$$\psi(\boldsymbol{X}) = \int \log\det(n\boldsymbol{I}(\boldsymbol{\beta}, \boldsymbol{X}) + \boldsymbol{R})p(\boldsymbol{\beta})d\boldsymbol{\beta} \tag{2.51}$$

where $n\boldsymbol{I}(\boldsymbol{\beta}, \boldsymbol{X})$ represents the Fisher information matrix. In case a linear model is considered, the Fisher information matrix does not depend on $\boldsymbol{\beta}$ and (2.51) is simplified [CV95]. While the non-Bayesian scheme maximizes the determinant of the Fisher information matrix (2.47), the Bayesian D-Optimal test design incorporates the prior assumption on the parameters. The result still is the maximization of the determinant of the Fisher information matrix but with an additional term, which originates from the prior distribution. The term $\boldsymbol{R}$ is the variance of the prior suggestion and is called the covariance matrix of the parameters. Solving (2.51) for a linear model leads to

$$\psi_{linear}(\boldsymbol{M}) = -\frac{k}{2}\log(2\pi) - \frac{k}{2} + \frac{1}{2}\log\det(\sigma^{-2}(\boldsymbol{M}\boldsymbol{M}^\top) + \boldsymbol{R})) \tag{2.52}$$

with the amount of parameters k and the design matrix $\boldsymbol{M}$. A Bayesian D-Optimal test design for a linear model can be found by maximizing (2.52), which can be simplified to

$$\boldsymbol{M}_{opt,D,Bayes} = \underset{\boldsymbol{M}}{\text{argmax}}(\det(\boldsymbol{M}^\top\boldsymbol{M} + \boldsymbol{R})). \tag{2.53}$$

The complete derivation of (2.53) and also other Bayesian optimal experimental designs, as e.g. Bayesian G-Optimal design, can be found in [CV95] for example. A utilization of Bayesian D-Optimal design in terms of MBC for a linear model is discussed and shown in [ZD11]. In that research, the Bayesian scheme is used to optimize the D-Optimal test design to incorporate space-filling test points by means of an adjusted prior covariance matrix. The disadvantage of non-Bayesian D-Optimal test design can be that many tests are executed at the input domain borders and none or just a few are planned space filling, which is addressed.

Considering nonlinear models, the integral in (2.51) has no closed solution and is of non-convex shape, which makes it hard to find a global optimum. One solution would be to put an assumption on the parameter vector $\boldsymbol{\beta}_0$ and find the best solution for $\psi(\boldsymbol{M})$ by considering the gradients of the nonlinear model [CV95]. The disadvantage of this method is the need for appropriate prior knowledge, since the guess about the parameter vector decides about the quality of the result. Other solutions are shown by numerical approximations of the integral, e.g. by Monte Carlo simulation, gridding methods [Rya+15] or Markov chain Monte Carlo simulation [CMP96].

2.2.3 Space-Filling Test Design

The space-filling test design methods try to reach an evenly distributed test point collection. They are not used to generate an optimal test design for a given model structure but more to cope with unknown input correlation. The basic idea is to generate a test design that gathers most possible knowledge in each region of the input domain. Typical low-discrepancy sequences are the Halton and Faure [War95; SBH10] as well as the Sobol sequence [Sob67]. The most utilized one is the Sobol sequence, which is calculable with low effort and creates a high-quality test design even in high dimensional space. The Sobol sequence counts as a quasi-random sequence and therefore shows a low discrepancy, which is a measure of uniform distribution. A low discrepancy is equivalent to a high space-filling quality. The Sobol sequence mainly consists of direction numbers for each dimension that are generated by a primitive polynomial and initial numbers and have to be defined. Each dimension has its own polynomial and direction numbers respectively. Given these direction numbers, the sequence in each dimension is calculated by an exclusive *or* operation of the preceding sequence value x_n and the direction number $v_{j(n)}$

$$x_{n+1} = v_{j(n)} \oplus x_n. \tag{2.54}$$

The direction number $v_{j(n)}$ is chosen from the binary representation of the sequence step n, where the rightmost zero entry counts. For instance, in sequence step $n = 7$ with binary representation $7_2 = 0111$ the fourth direction number is selected. The most popular guide on the implementation of a Sobol sequence is shown in [BF88].

Another method for the creation of low-discrepancy test designs is the iterative selection of candidates from a list that fulfills some distance criterion. A very popular one is the maximin criterion, presented by Johnson, et. al in 1990 [JMY90]. The first point selection is done at random or with a fixed selection and all further test points are added iteratively until the necessary amount of test points is reached. The selection criterion is based on the Euclidean distance, whereas the distance of all points belonging to the test set to all potential candidate points is determined. For all candidates, the closest test set point is selected, resulting in a distance vector for each potential new candidate. From the distance vector, the maximum value is determined, the corresponding candidate is selected and ultimately added to the test set. Given a candidate set $\boldsymbol{S}$ and a test point set $\boldsymbol{X}$, both of dimension dim, a new point is found by

$$\boldsymbol{x}' = \underset{\boldsymbol{S}}{\operatorname{argmax}} \min_{\boldsymbol{x}\in\boldsymbol{X}} d(\boldsymbol{x}, \boldsymbol{S}). \tag{2.55}$$

The maximin method creates very uniform test sets with low discrepancy but the result strongly depends on the initial selected test point. In case the candidate amount and distribution is sufficient, maximin test designs can result in a distribution with low input dimension specific density. This kind of test design can be useful, as it shows properties of an orthogonal test design, which is advantageous if each main effect needs to be recognized well. However, orthogonal test designs show poor behavior if one input has low or no effect on the output for example. In this case, a lot of measurements are taken without any process information gain. In experiments with non-discrete steps for each input variable, as they mainly take place in MBC, orthogonal test designs require a lot of measurements to identify each main effect. For these reasons, the Latin hypercube design (LHD) is introduced, which aims to find a design with test points evenly spread across each input dimension [SWN18]. The basic idea how to generate an LHD is to separate each input into discrete steps, with the input domain to be separated into hypercubes. As an example, if for each input 3 steps are assigned in a 2-dimensional input domain, it can be split into 9 squares. By permutation of the steps, an LHD can be found, where the design

$$\boldsymbol{X} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{bmatrix} \tag{2.56}$$

would be a valid design. It is easy to recognize that test points only placed at the principal diagonal is not a satisfying design. Other permutations can be found, but another criterion is needed to judge a good design. Therefore, the maximin and LHD is combined by Morris and Mitchell to use the strength of both test designs [MM95]. An initial LHD is created and optimized by a random selection exchange algorithm. Two values in only one randomly selected column are exchanged if the new design raises the maximin distance measure. Further, more efficient algorithms to create such designs are introduced e.g. in [JH08].

Space-filling test designs such as orthogonal, sequence-based, maximin, and LH designs or any combination of those can be applied to any type of regression model. Using polynomial models, the introduced optimal test designs are preferred if the model structure is known. However, in case no correlation of the input-output mapping is known in advance, the space-filling test designs are preferred.

2.2.4 Maximum Entropy Design

Another test design for discrete design spaces is based on the change of Shannon entropy. Originated in [Lin56], the idea is to maximize the information content of the test, based on the prior assumption of the parameters [SW87]. The entropy difference by selecting a test design $\boldsymbol{X}$ is defined by

$$\mathrm{Ent}(\Theta) - E_{\boldsymbol{y}}\{\mathrm{Ent}(\Theta|\boldsymbol{y},\boldsymbol{X})\} \tag{2.57}$$

with parameter space Θ and response $\boldsymbol{y}$. A test design contributes most to entropy change if the term $E_{\boldsymbol{y}}\{\mathrm{Ent}(\Theta|\boldsymbol{y},\boldsymbol{X})\}$ is minimized, which is mainly affected by the test design. If a model $\boldsymbol{y}$ with independent, identically distributed error ϵ is assumed, the entropy gain by a test can be expressed as

$$\mathrm{Ent}(\Theta) - E_{\boldsymbol{y}}\{\mathrm{Ent}(\Theta|\boldsymbol{y},\boldsymbol{X})\} = \mathrm{Ent}(\boldsymbol{y}|\boldsymbol{X}) - \mathrm{Ent}(\epsilon). \tag{2.58}$$

For linear models it can be shown that a maximum entropy test design is equal to a Bayesian D-Optimal test design [SW00] as expressed by (2.53). Independent if a linear or nonlinear model underlies, the maximum entropy design demands to solve an integral over the parameter space for the calculation of the entropy. The test design dependent term

$$\mathrm{Ent}(\boldsymbol{y}|\boldsymbol{X}) = -\int \log(p(y|\boldsymbol{X}))p(y|\boldsymbol{X})dy \tag{2.59}$$

contains the marginal probability distribution of the observation

$$p(y|\boldsymbol{X}) = \int p(y, \theta|\boldsymbol{X})p(\theta)d\theta \tag{2.60}$$

which is analytically intractable if the parameter space is continuous and not defined. A solution is to define different prior assumptions (K-point priors) and to put a weight on each assumption, which converts the integral into a calculable product [SW00]. In non-parametric models, as for example the Gaussian process model is, there is no need to solve the integral. The entropy of a multivariate Gaussian and a GPM respectively is defined by

$$\text{Ent}(\boldsymbol{y}_{GPM}|\boldsymbol{X}) = \text{Ent}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})) = \frac{1}{2}\log|\boldsymbol{K}| + \frac{\text{dim}}{2}(\log 2\pi e) \tag{2.61}$$

with dimension dim [CT91]. The entropy strongly depends on the covariance matrix $\boldsymbol{K}$, which in Gaussian process regression is adjusted to the data by the hyperparameter. Therefore, to generate a maximum entropy design for a GPM an assumption on the hyperparameter needs to be made, whereupon a test design by maximizing the entropy can be approximated. In case the input specific correlation lengths of a measurement noise free Gaussian process are equal, that is $\sigma_n = 0$ and $l_1 = l_2 = ... = l_i$ in (2.36), the maximum entropy design is similar to a space-filling distribution, with a more boundary focused design for increasing l_i [SWN18].

2.2.5 Test Design Conclusion

The most important test design methods are presented that match the introduced model types in section 2.1. The test design that should be applied depends on the experiment, knowledge, and model type. Once the model type is defined, the prior knowledge about the system to identify is crucial for the test design.

If no knowledge exists, an equal test distribution in the input domain, which is provided by a space-filling test design, is preferred. In case a polynomial model or Bayesian linear model should be applied and the maximum degree and input interactions are known, an optimal test design is the first choice. If additionally prior distributions for the parameters can be assumed, the Bayesian test design or the entropy-based test design should be the choice. More complex is the selection of the test design applying model types like neural networks or local linear model networks. Since the parameters at neural networks are hard to guess without any measurements, space-filling test designs are frequently applied. In LOLIMOT, it is possible to define the splits prior to the test and fix the model degree in each zone. With this assumption, an optimal test design is calculable. For HILOMOT models, the non-orthogonal splits are again hard to guess, which is why space-filling test designs are preferred. In GPM applications an entropy-based test design or space-filling test design is applicable. If prior knowledge about the hyperparameters is available, the entropy-based test design could be preferred, because it is calculable in reasonable accuracy and incorporates the input impact through the length scale.

The conclusion about the introduced test design mainly counts for tests that are designed prior any measurement. An iterative test design during testing yields more possibilities, because the guess about the non-known parameters can be updated after each test. This procedure and the test design potentially contributing the most is introduced in section 2.4.

2.3 Boundary Detection Methods

Contrary to computer experiments, combustion engine tests consist of critical boundaries where a test execution can potentially damage the engine. These boundaries are reached if input parameter combinations are applied that have an effect on different measurable physical quantities. A very simple example is the knocking and misfiring boundary in gasoline engines, which is mainly affected by the ignition timing. Very advanced ignition can cause strong knocking, which leads to high pressure oscillation in the burning chamber and therefore can damage the piston and piston rings. Retarding the ignition at some time leads to misfiring, where the ignition energy is not able to inflame the gasoline and air mixture anymore. This unburned mixture reaches the exhaust aftertreatment where it is burnt and releases high exothermic energy. The rise of temperature inside the catalyzer can cause damage to the substrate and catalytic material and consequently has to be prevented. This example is a very simple one-dimensional boundary problem. During engine calibration, the boundary problems arise as multi-dimensional, because interactions between input parameters cause non-orthogonal boundaries in the input domain. Further common physical boundaries to prevent damage at gasoline engines

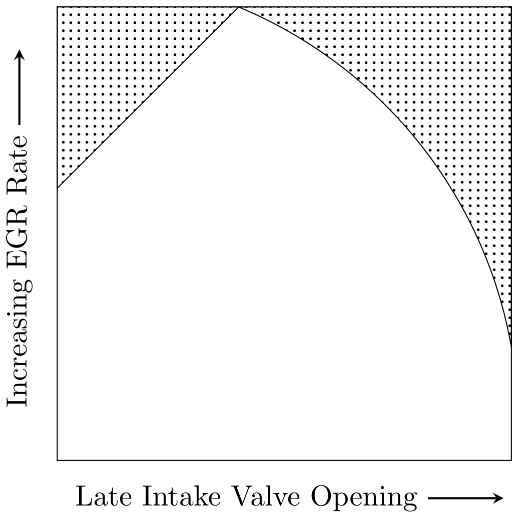


Figure 2.7: Schematic boundaries in a two dimensional input domain. The intake valve opening strongly interacts with the external EGR rate regarding non-drivable area. The amount of fresh air filling (upper right) and misfire (upper left) restrict the combinations.

are the maximum turbo charger turbine speed, temperature and pressure before turbo charger turbine, maximum pressure and temperature inside manifold, and the maximum peak pressure inside the combustion chamber.

An example for an interaction between two input parameters is given in figure 2.7. A variation of the intake valve timing and the external exhaust gas recirculation (EGR) rate is shown qualitatively for a constant cylinder fresh-air filling at partial engine load and for fixed exhaust valve timing. The marked area is a non-drivable area for two reasons. An early intake valve opening (Miller-cycle [EKP08]) results in a high amount of internal exhaust gas. Adding EGR is possible only in a low amount, since the burning stability is reduced if the amount of exhaust gas is increased, causing a misfire at a high exhaust gas amount. A very late intake valve opening (Atkinson-cycle [EKP08]) shows a low amount of internal exhaust gas. However, the intake valve keeps open during the compression stroke and fresh air is emitted. To achieve the same amount of cylinder fresh-air filling a higher degree of supercharging needs to be applied, which has some limit. Additional EGR reduces the fresh-air filling further more.

In diesel engine calibration, the amount of damaging parameter is nearly the same. Due to the different combustion principle, knocking does not occur and the exhaust temperature in general is lower. However, the maximum peak pressure, misfire, turbocharger speed, intake and exhaust gas temperature and pressure need to be monitored as well. Since the diesel engine is operable in a larger range, a popular method is to restrict the input domain by calibration target restrictions in the output domain. As an example, even if the target is not known in detail in advance of the test, the maximum allowed nitrogen oxide emissions could be applied as weak limit. Several of such targets can be used to restrict the modeling domain, leading to higher model quality at the same amount of measurements. Examples for the application of calibration targets and damaging limits can be found e.g. in [Mur+15] and [Cha+09] for diesel engine calibration and in [Dwy+13] and [WKO15] for gasoline engine calibration.

The engine boundaries, whether they are damaging or calibration target boundaries, need to be recognized during the test. The usual application of test points is a stepwise, vector-based adjustment of the parameters with boundary monitoring, see e.g. [SV17; Sta+13; Bau+13]. In case a limit is reached, a surrogate measurement is executed near the boundary. The test plan quality suffers from this condition because the optimality criterion is not met anymore if surrogate measurements are executed. One approach to hold the test design criterion is to identify the engine boundaries prior to the test design phase. Once the boundaries are known, it is possible to incorporate the restrictions during the test design phase and near-optimal solutions can be found. Furthermore the quality of boundary search is important for the engine model outcome. Data-driven models as introduced in section 2.1 have poor extrapolation behavior [BRL11] compared to physical models [Nel20]. Hence, the area to measure the training data in should be as large as possible. Different methods for boundary detection and boundary modeling to describe the gained knowledge are introduced in the following sections.

2.3.1 Manual Boundary Detection

A very simple method to find engine boundaries is a manual adjustment of each input parameter to its minimum and maximum value at different engine operating points. A practical solution is to stepwise increment or decrement the parameter value manually until the minimum or maximum

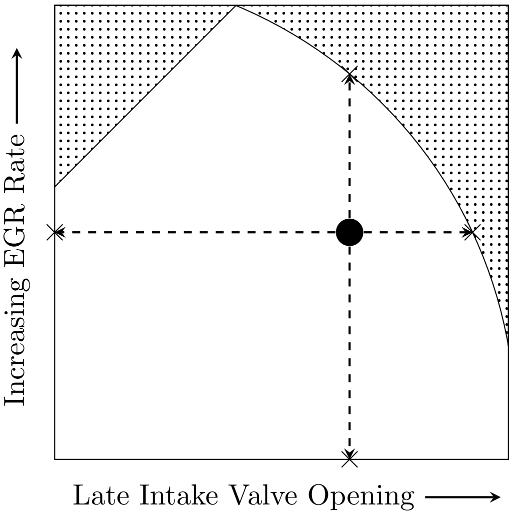


Figure 2.8: Boundary detection by single parameter minimum/maximum variation with given initial point. An exemplary solution for the example from figure 2.7 is shown.

value is reached or any limit is exceeded. To execute this procedure no automation system or complex algorithm is needed and it can be executed by each test bench operator, which are the main advantages. However, manual boundary detection has severe disadvantages. From a financial point of view, this procedure is very cost intensive because an unmanned operation during night or weekend is impossible. Another, more crucial disadvantage is the resulting boundary resolution.

Having a look at figure 2.8, the information about the boundary trend in this two-dimensional input domain is very low and strongly depends on the starting point. If only two inputs exist, one could say that manually more extreme combinations could be tested, but a vectorial setting of several inputs at the same time takes high effort and could end up very slow, because each step requires a sequential setting of all parameters. To overcome the complex measurement, reduce the test time, and allow unmanned operation, automatic boundary detection is introduced.

2.3.2 Automated Boundary Detection

Automated tests have the main advantage that it is possible to operate the boundary detection continuously night and day. Additionally, adaptive approaches can be applied to automatically involve already identified boundaries. Several different procedures are introduced in literature to identify the engine operable area. The most simple procedure is non-adaptive and executes a fixed star shaped search. Starting from a save base point, a central composite design (CCD) is used as target [Gsc+01; For+03], see figure 2.9 left. The vectorial setting starts from the base point and slowly approaches the particular test point with fixed step size. If a boundary is identified, the last valid step is used as boundary point and the base point is approached. Another non-adaptive solution is explained in [Mur+15]. Instead of a CCD design, a uniform design with a high amount of test points is calculated. The setting of the parameters is executed very fast just to judge if a limit occurs on the target vector or not. A near-boundary point is not identified but the result is a high amount of test points in operable and in non-operable areas.

Adaptive procedures always start with an initial measurement of several test points. The rapid hull determination (RHD) procedure utilizes the initial measurement information to calculate a convex hull [RA05]. The following search directions are the normal vectors of the hyperplanes, where the center of gravity of the particular facet is used as starting point. Figure 2.9 right shows this procedure exemplary for one iteration. Refer to section 2.3.3 for the explanation of the convex hull model. Compared to non-adaptive methods, the main advantages are the insensitivity regarding the starting point and the more rapidly growing detected area. A solution based on this method but which recognizes non-convex shaped test spaces is shown in [KSR06]. Further enhancements incorporate an estimation of the distance from the facet to the real boundary. The most promising test points can be selected from the estimation by the distance of the boundary to the hyperplane or by the amount of estimated obtained volume after measurement [RA05]. A search path optimization, based on the RHD methodology, to reduce the adjustment duration and a steady-state value estimation from the continuous adjustment is shown in [Yos+11].

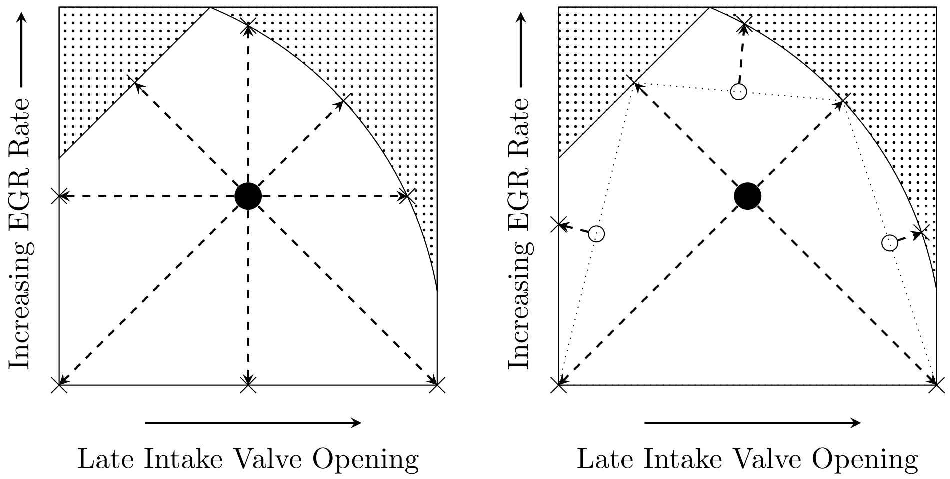


Figure 2.9: Left: Central composite design (CCD) boundary search as used in [Gsc+01] and [For+03]. Right: Rapid hull determination procedure (RHD) with one iteration and an input domain corner search as basic measurement [RA05].

2.3.3 Boundary Modeling

Once the boundaries and therewith the engine operable area is identified, the gained information needs to be processed. The most appropriate method to apply the boundary knowledge is the training of a surrogate boundary model, which is evaluable with any input combination. The surrogate model then can be used to calculate test designs as introduced in section 2.2.1 including the optimality criterion inside the operable area. A main differentiation about boundary modeling is the shape of the expected boundary. Convex and non-convex boundaries exist in practical applications and the resulting surrogate models are varying in their complexity and behavior. A hull is convex if any two inside lying points can be connected by a straight line without exiting the hull as is exemplarily shown in figure 2.10. Different types of both assumptions shall be introduced in the following sections.

Convex Boundary Modeling

The most applied and conventional boundary model is the convex hull model. The convex hull is a geometric model that is determinable without

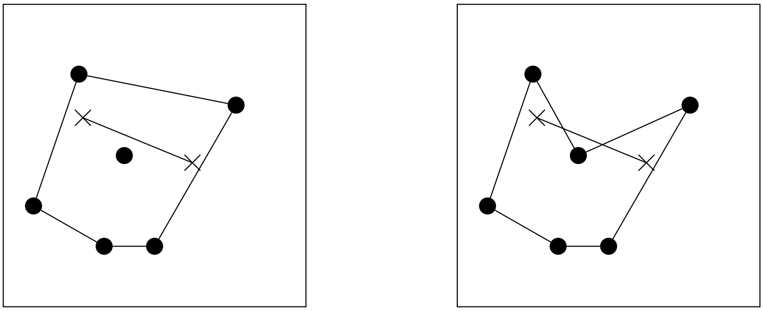


Figure 2.10: Left: Convex hull of a given point set. Each linear connection of two points inside the hull stays inside. Right: Non-convex hull of a given point set. Some linear connections of two points inside the hull exit the hull.

any parameter training. Two different modeling types can be utilized. The most convenient one is the calculation of all enfolding hyperplanes of a given point set. Given the normal vectors and orthogonal distances to the origin, the convex hull is defined by the Hesse normal form and any test point can be checked if it is inside the convex space by

$$\boldsymbol{A}\boldsymbol{x}^{\top} - \boldsymbol{b} \leq \boldsymbol{0} \tag{2.62}$$

for the normal vector matrix $\boldsymbol{A}$, the given test point $\boldsymbol{x}$ and the distances $\boldsymbol{b}$. An appropriate algorithm to determine the normal vectors and distances in a multidimensional domain is the quickhull algorithm [BDH96], which iteratively searches for the solution. Starting with an initial hull, in each iteration the furthest point is added and the hyperplanes are determined. For more information refer to [Bar19]. Another strictly convex method is to use the convex set of a point set to represent any point in the multidimensional domain. The convex set solution is defined by

$$\mathrm{co}(\boldsymbol{X}) = \left\{ \sum_{i=1}^{k} \alpha_i \boldsymbol{x}_i \,\middle|\, \sum_{i=1}^{k} \alpha_i = 1 \text{ and } 0 \leq \alpha_i \leq 1 \ \forall i \right\} \tag{2.63}$$

for a given point set $\boldsymbol{X}$. A given test point $\boldsymbol{v}$ is tested to be inside or outside the convex area by finding a solution to the α_i within the given constraints to represent $\boldsymbol{v}$. If no solution can be found, $\boldsymbol{v}$ is not part of $co(\boldsymbol{X})$. The search for a solution is challenging though and procedures were developed to reduce the testing effort. A utilization and comparison of a fast algorithm to the hyperplane-based convex hull model is introduced in chapter 3.

Non-Convex Boundary Modeling

In non-convex boundary modeling geometric procedures as well as probabilistic solutions are present. A simple, geometric approach is described in [Kow17], where a Delaunay triangulation of the point set is calculated. As boundary point labeled test points, which are inside the hull and not part of the convex hull border, are identified as non-convex areas. A vector starting at a central point to each non-convex boundary point is calculated and extended. All simplices that are beyond the boundary point are excluded from the triangulation. The new outer hyperplanes define the non-convex boundary model.

Another geometric non-convex hull is based on directed hypercones and introduced in [DKR17]. Either based on a central point or on a sequential construction from test point to test point, hypercones with a defined opening angle are designed that construct the non-convex hull. Similar to a convex hull, this approach enlarges the hull by additional measurements and defines non-measured regions as outside. However, the size of the hull strongly depends on the opening angle parameter that needs to be defined or optimized, and on the construction method itself.

A so-called “potato model” is first mentioned in [Knö+03] and well described in [Zag14]. It is based on a central point inside the hull and uses projections of measured hull points onto a unit sphere. The factor for extension or compression to reach the unit sphere by the vector from the central point towards the boundary point gives rise about the distance of the boundary from the unit sphere at a defined vector angle. The factor is a function of the Euclidean distances of the boundary points to the center and is therefore continuously available for each angle. Combined with a confidence term, unseen areas are interpreted as inside the hull and the hull shrinks towards measured boundaries. A radius needs to be defined that specifies the hull increase within unseen areas and which is a free parameter. This boundary model type is able to describe non-convex boundaries due to the exact incorporation of measured limit points. However, examples for two dimensions are given in [Zag14], but to the best of the author’s knowledge, a quality result for high dimensional input domains is not published.

The resulting hull strongly depends on the central point because the unit sphere is created on its base.

Non-geometric and non-convex approaches are compared to the convex hull approach in [Kie14]. Two classification models are mainly used for the comparison in terms of engine calibration. The prediction error variance (PEV)-based method is a probabilistic approach, using the variance of a given polynomial structure to determine the area where measurements are executed. The variance of a polynomial model only depends on the design matrix and the measurement noise. It is therefore independent of the output if the model structure is fixed. Additional to the model structure, the variance level needs to be defined at which a test point is found to be outside the measured area. The second approach used in [Kie14] is the support vector machine (SVM) model originating from the machine learning domain. An SVM tries to separate labeled data sets by a hyperplane with highest possible margin. Since a hyperplane is a linear separator, a kernel can be applied to separate the data in feature space and therefore achieve a nonlinear separation in original space. To train an SVM, the kernel for the mapping as well as a tuning parameter have to be identified, usually application-dependent by the user. In [Kie14] an approach for a leave-one-out-based optimization is investigated. The SVM approach, however, demands at least two labeled data sets. A similar but probabilistic-based classification is utilized in [Sch+15] in terms of engine calibration. The authors use a GPM classification model as described in [RW06] to discriminate operable and non-operable test points and to make predictions for unseen areas. Since it is developed in terms of an adaptive test design framework and is not focused on boundary modeling for a given consistent measurement data set, this method will be described in detail in section 2.4. Examples for the discussed non-convex boundary modeling methods for a given data set in a two-dimensional input domain are shown in figure 2.11.

Given a point set, there is one main difference between convex and non-convex boundary modeling. Due to the convexity definition, only one exact solution exists for convex boundary modeling if all points are part of the boundary. Contrarily, the hull precision in non-convex boundary modeling strongly depends on the boundary model. Except the Delaunay triangulation-based method (figure 2.11 (I)), the presented algorithms need to be parameterized and their quality therefore is strongly depen-

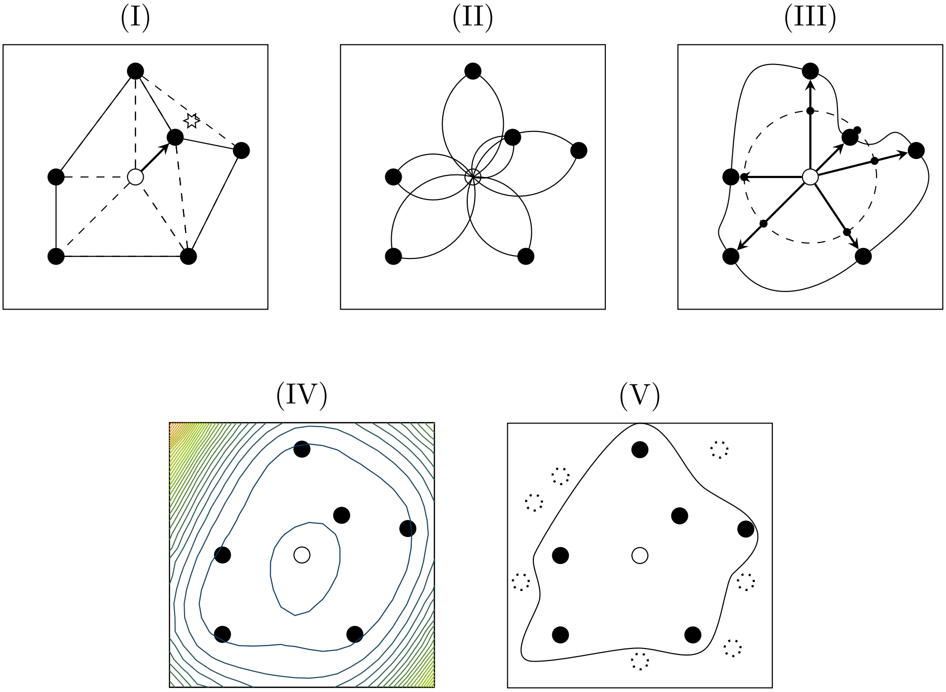


Figure 2.11: Exemplary representation of different non-convex hull description methods for the same data set. The circle without filling represents the base point, the filled circles are measured boundary points. (I): Concave hull representation based on a Delaunay triangulation [Kow17]. The asterisk marks the simplex which is erased. (II): Hypercone-based non-convex hull as introduced in [DKR17]. The central point construction method is shown with an opening angle of $\pi/4$. A point is inside the hull if it lies in any hypercone. (III): Potato model as described in [Zag14] with parameters $r_i = 0.9$ and $d_{max} = \sqrt{n/2}$. (IV): Prediction error variance (PEV) boundary model [Kie14] with quadratic polynomial model with interaction, linear term and constant. The level for discrimination needs to be defined use-case selectively. (V): Schematic representation of a support vector machine (SVM) model [Kie14] with discrimination in feature space. Since points of inside and outside label class are essential to train the model, arbitrary test points are represented by dotted points.

dent on the implementation. The main disadvantage of the triangulation is the performance: To calculate the hull, a Delaunay triangulation has to be performed, which effort and data storage is nonlinearly increasing with the dimension. The convex hull calculation, however, suffers from the same property but at a significantly lower level. In terms of MBC, input domains up to 12 dimensions and 2000 test points are applied, where hyperplane-based methods collapse without the utilization of a computer cluster.

2.3.4 Boundary Incorporation during Test Design

In section 2.2, different test design strategies are shown to maximize the information content of each single measurement regarding model quality. However, the test design strongly depends on prior assumptions about the design space regarding engine boundaries and physical actuator boundaries. If any information about the boundaries is available, it can be incorporated into the test design. Possible information could be boundary detection measurements and known parameter input restrictions with or without interactions or simply non-relevant input domain areas. The consideration of the boundaries is necessary because each deviation of measured and desired test points could lead to a decrease in the resulting engine model quality.

In practical applications, two different characteristic types of MBC test designs exist. The classic approach is to measure the hypercube input domain with restrictions regarding input and output limits. Another approach is executed if a good base calibration already exists but needs to be refined. The so-called variation around the base aims to allow variations only slightly around a given base map in the engine speed and load domain. The most important fact with this design is that for each speed and load grid point a different minimum and maximum value for each input exists. The upper and lower limits are commonly applied as maps for each parameter [Bau+13; Bol+13]. An example for a variation around the base for the fuel rail pressure is shown in figure 2.12.

Common commercial software solutions to define and calculate an MBC test design, such as IAV Kasai [IAV20] or ETAS ASCMO [ETA20a], include 4 different types of input constraints:

- *Map-based constraints* define a lower and upper boundary for each engine speed and load grid point.

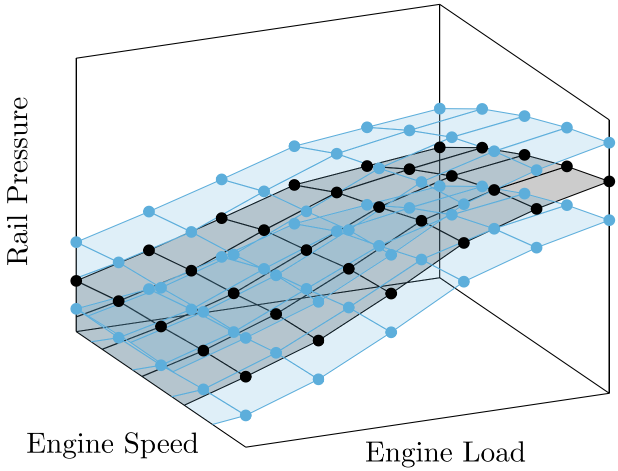


Figure 2.12: Map-based constraint example for the fuel rail pressure. The base map is shown in black. The blue surfaces map the upper and lower limits and constrain the design space dependent on engine speed and load.

- *Table-based constraints* define a lower and upper boundary for selected engine speed and load grid points. Interpolation is done linearly. The main difference to a map-based constraint is the way it is defined, because there is no need to define a lower and upper boundary for each speed and load grid point.

- *Inequation constraints* flexibly define any type of boundary with interaction between any amount of input parameter. They are hard to define, but also complex interactions can be incorporated in order to prevent overlapping of multiple injections for example.

- A *boundary model constraint* can incorporate boundary search results as introduced in the preceding sections.

While table- and map-based constraints only apply for a global DoE including engine speed and load, inequations and boundary models are also applied at local DoEs with only one operating point. It mainly depends on the way the test design is computed how those constraints are incorporated. The easiest but also very fast solution is a discretization by a low-discrepancy candidate set as by a Sobol sequence for example. The sequence is executed iteratively and each sequence member that does not fulfill the constraints is rejected until a defined amount of candidates is generated. The first n sequence members could be applied as a low-discrepancy test plan directly. To find a test design which fulfills other criteria, like the maximum entropy criterion for example, an exhaustive search or a greedy algorithm is applicable using the generated candidates. Another solution without a candidate set would be a gradient descent optimization (see e.g. [CT17]) with nonlinear constraints. Compared to a greedy algorithm, this way is much more exhaustive in case the amount of candidates is low and also the risk of ending up in local optima is higher. A global optimization, as e.g. a particle swarm optimization introduced in [EK95] or a genetic algorithm (refer to [Kra17]), could be applied, which again is much more exhaustive than a local, gradient-based algorithm. However, with a global optimization the highest test design quality can be achieved because the global optimum can be found without a limiting discretization.

2.4 Adaptive Test Design

The adaptive test design, also called active learning procedure, has a longer history especially in classification tasks as speech recognition or image classification. The aim here is to selectively use data from a data stream or database to feed the model with. The main problem is to generate the labeled data, where often human interaction is necessary. Therefore, active learning is used to minimize the labeling effort. A good overview of several methods is given in [Set09]. Although the focus is on classification model training, regression problems are very similar and the methods can be adapted to it. However, in MBC, the requirements are a bit different and the development history needs to be considered.

The evolution of MBC can be divided into three steps. When MBC was introduced, boundaries where mainly not considered and all tests were designed in a user-defined input domain. The first main advancement was the introduction of a preceding boundary detection and the utilization of boundary models, where the system identification measurements are executed within the modeled boundaries. An obvious enhancement of this process is the adaptive test design. Instead of planning all measurements in advance of the test, information from already taken measurements can be used for subsequent tests during the measurement campaign. Several procedures exist with different focuses. On the one hand, methods are developed to incorporate the boundary identification into the system identification process. Using an adaptive test design framework, subsequent tests can be planned with a focus on model quality on the other hand. The most important adaptive test design procedures shall be introduced in the following. For a better understanding, a classification of the methods and their process is defined as follows. An overview is given in table 2.2.

Table 2.2: Classification of non-adaptive and adaptive test design strategies in terms of MBC

Name	Fixed Design Boundary Search	Adaptive Boundary Search	Adaptive Model Quality Improvement
1-Stage Offline	-	-	-
2-Stage Offline	x	-	-
1-Stage Half Adaptive (Focus Boundaries)	-	x	-
1-Stage Half Adaptive (Focus Model Quality)	-	-	x
2-Stage Half Adaptive	x	-	x
2-Stage Full Adaptive	-	x	x
1-Stage Full Adaptive	-	x	x

- *1-Stage Offline:* Pre-calculated test design, which is measured without any adaptation.
- *2-Stage Offline:* Boundary finding in the first stage and measurement of a pre-calculated test design with boundary knowledge incorporation in the second stage.
- *1-Stage Half Adaptive (Focus Boundaries):* Adaptive boundary search with pre-calculated test design. Points that cannot be applied are e.g. skipped.
- *1-Stage Half Adaptive (Focus Model Quality):* Adaptive test design only focusing model quality improvement, while boundaries are not considered.

- *2-Stage Half Adaptive:* Measurement of fixed boundary search design in first stage with subsequent adaptive test design that focuses model quality improvement considering first stage boundaries.
- *2-Stage Full Adaptive:* Adaptive boundary search in first stage and adaptive test design focusing model quality improvement in a subsequent stage considering first stage boundaries.
- *1-Stage Full Adaptive:* Joint adaptive test design incorporating boundary search and model quality improvement as combined criterion.

The existing adaptive test design methods are additionally subdivided by their focus. The next section deals with an introduction of boundary search focusing methods. Since the most adaptive procedures focus on the model quality, the following sections are structured by the applied model type dealing with local model network, neural network, and GPM-based procedures. Each introduced method is additionally classified by the defined categories.

2.4.1 Boundary Search Focusing Methods

Different to the introduced methods for boundary detection in section 2.3.2, the boundaries can be considered during an adaptive test design with the focus on generating measurements to train a specific model architecture. The objective of this method is to achieve a similar test point distribution as if the boundaries would be known in advance. The most simple method is to run an initial test design [Sam09] and use its measurement to calculate a boundary model for the subsequent test design. This procedure is independent of the model type because each test design criterion is applicable as in a non-adaptive approach. The method is equivalent to an automatized *2-Stage Offline* process.

An extension is the calculation of test points iteratively with an update of the boundary model after each measurement. A procedure following this approach with the utilization of a discriminating hyperellipsoid is shown in [BGC07]. Unfortunately there is no detailed description neither of the boundary model calculation nor of the boundary exploration strategy. From a general methodological point of view, this adaptive test design also does not exploit the beneficial possibilities it offers. However, since it utilizes a boundary model that is iteratively updated and a test design that is adaptively planned, the method is classified as *1-Stage Full Adaptive*.

2.4.2 Local Model Network-Based Methods

An adaptive test design approach based on a LOLIMOT model is introduced in [Kow17]. The main focus of this method is the identification of the model parameters. The input domain is defined by a non-convex hull model as shown in figure 2.11 (I), which is found in a first step by a fixed initial measurement. In the main measurement procedure, an iterative test design is applied, where a set of test points is calculated in each iteration. Different criteria are applied, depending on the change of model structure (i.e. number of splits and local polynomial order) during the last iteration. In case the model structure changed, the test design criteria for the subsequent iteration focuses on generating information for the model structure. This is done by a combination of space-filling test points and a criterion called model quality decrease. A model committee is used to identify test points that probably result in a model quality decrease. The committee provides an estimated measurement by averaging the output of all belonging models. This estimation is applied to train the model in charge and the quality change can be observed. It is assumed that a non-matching model structure is the reason for the quality decrease and therefore this test design method is applied where the structure still changes.

In case the model structure remained constant during the last iteration, criteria focusing the local model quality are applied. If an optimization criterion is defined, the candidate nearest to the estimated optimum is added to the test design. Additionally, test points that lead to the highest model quality increase and the highest model parameter change are added for the subsequent iteration. The base for these estimations is again the model committee to generate the estimated measurement output for each candidate. This adaptive procedure can be classified as *2-Stage Half Adaptive*.

An enhancement of the LOLIMOT-based approach is an adaptive test design with using axis-oblique partitioning. The axis-orthogonal splitting procedure suffers from nonlinear processes with significant change in diagonal input domain directions. Also high-dimensional modeling is considerably improved by axis-oblique partitioning, as shown in section 2.1.3. Adaptive test design procedures using axis-oblique partitioning with tree-based splitting are introduced e.g. in [HJ11; Sta+13; HN13; Har14]. The most simple procedure is shown in [HN13], called HilomotDoE. The worst local model is determined where both the bias and variance error is considered. A next test point is then planned space filling in the worst partition.

A more complex procedure is shown in [Har14] where a method called bagging [HTF09] is used. Therefore, several data sets are randomly created from the collected data and a HILOMOT model is trained for each data set. The resulting model is built by averaging the output over all trained models. A next test point is placed at the location with highest variance of the bagging model. This procedure mainly generates test points in strong nonlinear areas but has the disadvantage that the computational effort is very high to generate the different models. These methods do not incorporate a boundary search, which means the boundaries need to be investigated prior to the adaptive test design measurements. Therefore, these procedures are classified as *2-Stage Half Adaptive*.

Another solution is investigated in [Sta+13] called custom output range (COR), using an axis-oblique local model network as described in [HJ11]. The main differences to HilomotDoE are the recognition of engine boundaries and a relevant input domain as well as the test design method. At first, a fixed initial test design is executed that provides the measurement for the drivable input domain and to train the first models. A convex hull is applied as boundary model for the subsequent iterative test design. The input domain is further restricted by the COR methodology, which puts a constraint on the current model outputs respectively. Candidates in the input domain leading to estimated output values outside a defined range are not taken into account during each iteration. The desired effect is to identify the system only in calibration-relevant areas and therefore reduce the amount of necessary measurements. Within this input area, a next test point is planned by a modified space-filling criterion. The maximin procedure is applied in a combined input-output domain, resulting in a more dense test point distribution in nonlinear model output regions. Another possibility is to apply the maximin procedure only in the output domain, as proposed in [Sta+14]. Compared to the test design in the combined domain, this approach leads to a similar result in a presented theoretical example with strong nonlinearity. This procedure offers a combined approach for boundary search and model quality improvement and is therefore classified as *1-Stage Full Adaptive*.

2.4.3 Neural Network-Based Methods

The applications of neural networks for an adaptive test design in terms of MBC are rarely present. Only little literature is available where MLP or RBF networks are used. Since local model networks outperform classic neural networks especially in terms of interpretability, uncertainty representation and training speed, they are rarely considered for an application at the test bench. However, the first outstanding application of an adaptive test design procedure with the scope of MBC was introduced in [Pol+03] and utilizes MLP networks among others. The main idea is to create the model variance with a model committee, which firstly was investigated in [SOS92]. This so-called query by committee (QBC) method is applied in the mbminimize algorithm that implements MLP networks and linear models like polynomial models for the variance calculation. A next test point is added at the input domain location with highest deviation of the model committee outputs. Since this procedure could lead to several measurements in the same region, a confidence term, which penalizes close test points, is introduced in [Sun+07] to stabilize the QBC algorithm. Additionally a weighting is applied that allows finding optimal test points for the optimization of several outputs with different importance. A solution for the incorporation of engine boundaries within the mbminimize algorithm is given in [Knö+03]. The test design does not focus the boundary identification but a limit model is incorporated to restrict the input domain for the further test point search. Since the search space should not be restricted excessively, different approaches are proposed like the "potato model", further confidence-based models, output-driven regression models as used in the COR methodology, and also geometric models. An evaluation of the different model types is not given, which is why they are not explained here further. The utilization of a boundary model with a test design focusing model quality improvement allows this methodology to be classified as *1-Stage Full Adaptive.*

Another procedure shall be presented whose specialization is not a test design for model quality improvement but rather one to investigate the boundaries. The argumentation of [SV17] is the occurrence of calibration optima mostly near boundaries. Hence, the best model would end up in a good description of the measured area, but if the optimum is not part of it the best solution will not be found. For that reason, the authors use an iterative test design with a simple space-filling design in a modeled boundary area. The boundary model in this case is comparable to the COR methodology with a response model for each limit parameter.

These models are used to generate information back in the input domain to constrain the input-driven space-filling design. The maximin solution to generate the space-filling design in this case is not driven by a fixed candidate set but the authors use a nonlinear optimization to place new test points during each iteration. Although the test design is weak regarding model quality improvement and does not exploit its possibilities in terms of output behavior incorporation, this procedure is classified as *1-Stage Full Adaptive* because it updates the space-filling test design in each iteration with adaptive boundary knowledge incorporation.

2.4.4 Gaussian Process-Based Methods

The usage of GPM in adaptive test design strategies has great acceptance. Thus many different methods exist for a selective test point choice. The well uncertainty estimation in GP regression provides a good base to design algorithms, which aim to iteratively optimize the overall or regional model quality. Several algorithms are introduced in non-MBC applications and some of them are also investigated with focus on MBC and combined with boundary modeling. Present methods in non-MBC and in MBC applications shall be shown. Similar to an optimal test design for polynomial models with a given structure, the test point location in the input domain can be chosen to reduce the estimated uncertainty of a GPM. During online modeling, the GPM is updated after each measurement, which is why the uncertainty estimation changes dramatically. Typical procedures aim to reduce the mean variance by searching for a test point that reduces the integrated mean squared error (IMSE), or to reduce the maximum variance by the examination of the maximum mean squared error (MMSE) [Sac+89].

The reduction of the maximum mean squared error is focused in [Gen+15] in a sensor calibration problem. In each iteration, the sequential test design adds the candidate to the design that minimizes the maximum variance of the GPM if added to the covariance matrix. The maximum variance is detected by a predefined grid at which the variance is evaluated and the maximum value observed. The test point search is done with a genetic optimization algorithm to find the global optimal candidate to be added to the test design. In each iteration of the optimizer, the inversion of the covariance matrix of training points $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ has to be done (refer to (2.38)), which makes it computationally expensive for a high amount of training points. Since the use case here is a sensor calibration with 3 inputs, the amount of maximum training data is low and also no boundary search is necessary. The use of maximum variance in a different way is used in an MBC online optimization framework in [Ber12]. Subsequent to an initial measurement phase a next test point is added to a sequential design that shows the highest prediction variance. No further explanation about the search for the next test point is made. However, since the maximum variance is not determined after the test point is added but the one is used which shows maximum uncertainty prior measurement, the search is less expensive than the method by [Gen+15]. Engine boundaries are not considered in this framework, for which reason it is classified as *1-Stage Half Adaptive (Focus Model Quality)*.

A good solution to calculate the IMSE is presented in [BP15]. The authors show an analytic solution to the integration over the input domain for GPM with squared exponential covariance function. Additionally, a combined criterion for IMSE and MMSE is shown to enhance the optimization stability, because the IMSE function shows rough behavior with several local minima. The use case, at which this solution is experienced, is a physical model of a compression spring force with 4 varied geometrical parameters. Next to minimum and maximum values of the inputs, further boundaries cannot be considered in this procedure because the analytic integration only works for the whole input domain, meaning a hypercube. This methodology can hardly be applied to MBC without the integration of a boundary model to the calculation.

Methods that are based on the predictive information content of subsequent measurements seem to be very beneficial. The most obvious way is to maximize the entropy, which in GPM only depends on the covariance matrix as shown in (2.61). Since the model can be re-trained during the adaptive test design based on the new measurements, the hyperparameters of the GPM change in each iteration. With changing hyperparameters the covariance matrix entries vary for which reason it makes a difference to calculate an entropy-based test design during or in advance of the tests. In [Rai15] an iterative entropy-based test design is simulatively compared to space-filling and maximum variance-based design strategies for different process shapes. No clear advantage of any of the strategies is exhibited and therefore a change of the test design criterion during the measurement is developed. It is shown that for different process shapes also different test design strategies are necessary, which makes it hard to automatize.

Entropy-based test designs often show the disadvantage that they overestimate the information content in near boundary areas [GKS05]. To overcome this problem, mutual information (MI)-based test designs are used. The calculation of the MI between already seen and non-seen areas can be considered as to penalize the entropy near already taken measurements. This procedure is introduced in optimal sensor-placement problems [GKS05], used in people-tracking problems [ZKN12] and also transferred to MBC with several outputs but with a low amount of inputs [Xie16]. The MI criterion for a Gaussian random variable Y, an input domain discretization candidate set $\boldsymbol{S}$ and test points $\boldsymbol{X}$ can be formulated by

$$\mathcal{I}(\boldsymbol{X};\boldsymbol{S}\setminus\boldsymbol{X}) = \mathrm{Ent}(Y|\boldsymbol{X}) - \mathrm{Ent}(Y|\boldsymbol{S}\setminus\boldsymbol{X}). \tag{2.64}$$

With setting $\bar{\boldsymbol{X}} = \boldsymbol{S}\setminus\boldsymbol{X}$ the MI for a test point $\boldsymbol{x}^*$ given a trained GPM is calculated by

$$\mathcal{I}(\boldsymbol{x}^*) = \frac{\sigma^2_{\boldsymbol{x}^*} - \boldsymbol{K}_{\boldsymbol{x}^*\boldsymbol{X}}\boldsymbol{K}^{-1}_{\boldsymbol{X}\boldsymbol{X}}\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{x}^*}}{\sigma^2_{\boldsymbol{x}^*} - \boldsymbol{K}_{\boldsymbol{x}^*\bar{\boldsymbol{X}}}\boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{x}^*}}. \tag{2.65}$$

If a candidate set $\boldsymbol{S}$ is defined, as e.g. a grid or a Sobol sequence, a next test point can be found by determining the candidate that maximizes $\mathcal{I}(\boldsymbol{x}^*)$ [GKS05]. To solve (2.65), however, is computationally expensive if a large number of candidates is present, because the matrix $\boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}$ needs to be inverted, which has size $(n_{\text{cand}} - n_{\text{meas}}) \times (n_{\text{cand}} - n_{\text{meas}})$. Since the input domain volume grows exponentially with the dimension, the amount of candidates for a similar discretization quality needs to grow with the same amount. Even though the matrix inversion only needs to be done once per test point search, the covariance matrix multiplication in the denominator is also dependent on the candidate set size and is performed for each potential next test point. For these reasons, the application of the MI criterion in literature raises from spatial problems and is applied in MBC only in low-dimensional calibration work. The MBC use case in [Xie16] starts with an initial test design that necessarily has to cover the whole input domain, because boundary finding methods are not considered. This procedure is classified as *1-Stage Half Adaptive (Focus Model Quality).*

The presented procedures using GP regression models do not take engine boundaries into account. Two further solutions shall be presented, which aim for MBC and consider the exploration of the engine operating space additionally. The presented method in [The+16] actually is not developed for MBC at an engine test bench but rather to optimize drivability investigations with a vehicle. The limitations in output space are defined by calibration constraints rather than engine or vehicle damaging limits. The presented algorithm is applicable to engine test bench experiments anyway. The procedure consists of three steps, starting with an initial measurement campaign as usual. The second phase aims at the exploration of the input domain. Test points are iteratively calculated by means of a D-Optimal test design assuming a linear model. This type of test design focuses the boundaries and therefore calculates test points mainly at the borders of the input domain. The boundary model is constructed by a combination of the models and a distance criterion. For each output a model is calculated. Input combinations leading to values outside the boundaries in the output domain are not considered. Additionally, a distance criterion to already measured test points in the input domain is defined. Test points which are far away from already taken measurements are also refused. The restrictive behavior of the combination of both criteria is described to be tunable. By means of this adjustment possibility a more restrictive and therefore exhaustive search or a more risky search with a high likelihood of limit violations is executable. The third phase simply adds space-filling test points to the experienced measurements within the boundaries, defined by the boundary model. Even though three phases are described, the initial test phase is necessary as start for any type of adaptive procedure. However, the second phase is solely used as boundary search and the last phase as model quality improvement phase. Since there is a switching criterion between a boundary search and model quality improvement phase, this procedure is classified as *2-Stage Full Adaptive*.

Another procedure also uses output modeling to incorporate boundaries but in a different way than all previously described methods. A first, probability-based method is discussed in [Sui+15] regarding safe online optimization and is explored in terms of movie recommendation and a therapeutic example. This method aims to find the maximum function value but without falling below a lower limit during exploration. However, more interesting in terms of MBC is the algorithm introduced in [Sch+15] that is based on a similar boundary estimation procedure. A combined binary classification and regression method based on a Gaussian process is used during an iterative test design. Therefore the Gaussian likelihood function is set-up as a product of a regression and a classification likelihood. It is possible to approach both labeled data and continuous data with this solution. Labeled data is generated by means of the test bench automation, which provides the information about drivability after each test point is tried to be set. Continuous data is gathered by the measurement of each limit channel. However, due to the classification likelihood that is non-Gaussian, an analytic solution for the GPM posterior and marginal likelihood does not exist. A computational expensive approximation has to be performed, which here is based on the Laplace approximation. Unfortunately the hyperparameter selection by marginal likelihood optimization as shown in section 2.1.4 is much more exhaustive than in GP regression [RW06]. For this reason, the hyperparameters for the boundary model are suggested to be fixed and not varied during the adaptive test design. For the implementation in an adaptive test design, the drivable operating range is examined by the mean value of the boundary model μ_{g*}, where the prediction variance σ_{g*} is used to lower the probability of a misprediction for drivable test points, given a confidence term ν to regulate the "safeness" behavior

$$\mu_{g_*} - \nu\sigma_{g_*} \geq 0. \tag{2.66}$$

In this definition, a positive-labeled test point is defined as inside the boundary where a negative value means non-drivability. In each iteration, (2.66) is applied as constraint for a variance-based optimization where a test point is determined that shows highest prediction uncertainty. A practical example at an engine test bench, only to show the performance of the boundary model, is given in [Har+16]. The classification model itself is trained with labeled data, i.e. after the approach and measurement each point is specified as drivable or non-drivable by the automation system. A pre-defined test plan is used as test point selection base and in each iteration the next test point is estimated by the model to be drivable or not. In case it is assumed to be non-drivable, it is simply skipped.

A more relevant practical application using the same boundary model but additionally the iterative, variance-based test design, is given in [Sch+17]. Unfortunately, both examples mainly consider the feature of not hitting a boundary during the measurement phase but do not compare the applied methods to a classic, non-adaptive approach in terms of model accuracy or measurement duration. In the second part of [Sch+17], an extension is proposed, where not only labeled data is used for the boundary model but the continuous data of an output serves as training data for a discriminative model. The authors argue that this leads to a better prediction of the boundary and therefore less limit violations during the measurement. The regression output is mapped to a unit interval by a risk function, which also defines the mapping shape and therefore the conservatism of domain exploration. In [Sch+18] it is mentioned that also several outputs can be combined to one discriminative model by the risk function, but a procedure is not given. Several solutions could be applied, for example to take the maximum risk over all outputs as training input. A comparison with the binary classification model by [Har+16] is given, which on the one hand shows a safer prediction of the boundary but on the other hand does not offer an improvement of model quality.

Another benefit of the discriminative model is to overcome the issue of hyperparameter estimation in GP classification models, because a GP regression model is trained [Sch+18] and the marginal likelihood can be solved without approximation. The iterative online training of hyperparameters of the boundary model reduces the amount of pre-knowledge necessary for the process parameterization. During the measurement, the missing expert knowledge is learned by optimization and simplifies the application without loss of quality. Whichever boundary model is applied, the safe active learning called procedure is a *1-Stage Full Adaptive* approach.

2.4.5 Method Summary

The preceding sections introduced several strategies with very different scopes and implementations. A summary of all methods, their classification, and specific realization details is shown in table 2.3. The amount of GPM-based strategies is noticeable. The diversity of all methods in terms of general procedure, boundary model type, and test design criterion is remarkably high, which makes it hard to guess what combination should be applied. To judge the efficiency of a procedure, evaluation criteria and process requirements have to be established, which is part of the subsequent section.

Table 2.3: MBC adaptive test design procedure overview

Method	Classification	Regression Model	Boundary Model	Test Design Criterion
[Kow17]	*2-Stage Half Adaptive*	LOLIMOT	Concave hull	Space filling / Model quality decrease / Model quality increase / Model parameter change / Optimization criterion optimum
[HN13] [Har14]	*2-Stage Half Adaptive*	HILOMOT	Not specified	Local space filling / Model committee dissent (QBC)
[Sta+13]	*1-Stage Full Adaptive*	Axis-oblique local model network	Convex hull / Output-driven boundary model	Space filling in input-output domain
[Pol+03] [Sun+07] [Knö+03]	*1-Stage Full Adaptive*	MLP network	“Potato Model”	Model committee dissent (QBC)
[SV17]	*1-Stage Full Adaptive*	Neural network with linear base functions	Output-driven boundary model	Space filling

Continued on next page

Table 2.3: MBC adaptive test design procedure overview *(Continued)*

Method	Classification	Regression Model	Boundary Model	Test Design Criterion
[Gen+15]	*1-Stage Half Adaptive (Focus Model Quality)*	GPM	no model	Maximum variance prior measurement
[Xie16]	*1-Stage Half Adaptive (Focus Model Quality)*	GPM	no model	Mutual information (MI)
[The+16]	*2-Stage Full Adaptive*	GPM	Output-driven boundary model / Maximum input distance	D-Optimal / Space filling
[Sch+15]	*1-Stage Full Adaptive*	GPM	Output driven (Combined GP classification and regression)	Maximum variance
[Sch+18]	*1-Stage Full Adaptive*	GPM	Output driven (GP regression)	Maximum variance

2.5 Open Questions and Research Topics

It is reasonable that an adaptive test design approach is able to excessively raise the model quality given a defined amount of measurements. However, the introduced methods differ in their target regarding engine boundary identification, applied model type and model error decrease and its assessment. Next to these indicators, another important requirement to the MBC process and its methods is the difficulty of application. If a lot of parameterization tasks have to be done with the requirement of methodological expert knowledge, a calibration engineer will have trouble with the exercise. Another demand is the applicability to many different measurement tasks. Due to very diverse engine configurations and also calibration tasks, the same methodology needs to be reasonable for different shapes of input domains and output behavior. A boundary search therefore is indispensable. Following these requirements, a *1-Stage Full Adaptive* process seems to be the best solution, because a 2-stage approach always needs to have a switch criterion, which is hard to automatize without parameterization.

As acquired in section 2.1.5 and also the amount of different adaptive test design strategies indicates, the GPM is the first choice for an MBC task. Within this research the GPM with cross-validation optimization training is applied. The most critical property of this model class in an adaptive test design process is the training effort, which needs to be considered accurately. In case the model training takes too long and no alternative solution is given to calculate a new test point, the test duration will increase and the quality per test time ratio is lowered. Only two solutions for a *1-Stage Full Adaptive* process with GPM are known, which are related and were presented in the preceding section. However, several topics still are not covered by these processes and need to be investigated.

<u>Piecewise-defined limit behavior</u> is a topic that the process deals with in case the classification GPM is used. With the regression discrimination model, it is nearly impossible to reproduce piecewise-defined output functions that exhibit a constant value in wide ranges and a sharp increase near the operable area boundary. One example would be the misfire behavior. Its value is specified as the rate of misfire events that happened during the past combustion cycles. During most operation it has value 0 because each cycle burns well and one always tries to avoid misfire. In case a misfiring happens, either the automation system or the test bench operator

tries to get rid of this operation status, because it likely can damage the catalyst due to high exothermic energy and therewith a high temperature raise. Measurements with misfire values unequal to 0 are therefore very unlikely and are tried to be prohibited by operation. As an example function consider

$$f(x) = \begin{cases} 1 & x < -20 \\ -0.1x - 1 & -20 \leq x \leq -10 \\ 0 & x > -10 \end{cases} \tag{2.67}$$

with x representing the spark timing in advance of the top dead center. A function value of 1 equals 100 % misfire. A test bench measurement with lowest spark timing might be done at $x = -10$, lower spark timing values would be applied during test point setting only temporarily. For such an approached test point the classification GPM knows that it is part of the non-operable area due to its label. A regression model only experiences the mean measurement of the surrogate test point, which has value 0 though. The classification GPM however does not offer hyperparameter training, which is essentially required for an easy to apply procedure.

The optimal adaptive test design strategy for an MBC application with a GPM is, to the best of the author's knowledge, not present in literature. Many different test design strategies are proposed to achieve a maximum knowledge gain for a specific model. A comparison of entropy and variance-based test design with a non-adaptive low-discrepancy one is given in [Rai15] in terms of a GPM application. No clear solution could be discovered for an easy to apply strategy and besides these test designs others are possible. The question about the optimal adaptive test design for a universal MBC application therefore still is not clarified. The type of test design naturally depends on the problem. In MBC the most often used model quality criterion yet is the generalization error, defined by validation measurements evenly spread over the input domain. A best-matching test design approach has to be found that satisfies different process behaviors, which is rated in terms of generalization error.

More than one output is a common use case in almost each calibration task. All proposed methods, both within GPM designs and also other model class-specific test design strategies, do not take into consideration a test design matching several outputs. In literature, some proposals are made like a round-robin approach [Klö09] or a batch optimization [Kle+13]. With a batch procedure, where a test design focuses only one output until a criterion is reached, an automatic switch has to be developed or calibrated. In addition, this procedure is not a multicriterial test design optimization, because only one output is taken into consideration at a time. A round-robin procedure, that calculates a single test point for each output without considering the other outputs, is not multicriterial as well and suffers from a uniform weighting of each output, where the model quality is not considered. Hence, an optimization strategy for several outputs has to be investigated.

A continuous measured input domain volume increase is another very important property of a full adaptive test design strategy. Many of the adaptive test designs do not consider an extension of the measured area. The trained models should only be examined within their measured area. An optimal calibration solution outside this area hence would be not recognized. The known *1-Stage Full Adaptive* solutions do consider engine boundaries but do not consider how to enlarge them consequently while at the same time optimizing the model quality. A strategy for this has to be developed and is one of the research topics.

The criteria to rate an MBC methodology have to be defined to judge the quality of a developed process. The main goals of a measurement result are defined by

- measured input domain size,
- average model quality,
- test duration.

In high dimensional input domain problems, it is hard to judge the measured domain size. Especially when the measurable area is non-convex, there is no criterion to identify the covered area easily. As long as it is convex, the best solution is to consult the convex hull volume, which is calculable in a post-processing step. Another reason to use this volume as criterion is that often the convex hull is used [IAV20; Bau+13; SI09] to prevent model extrapolation and therefore it is one decisive measure for the subsequent optimization quality.

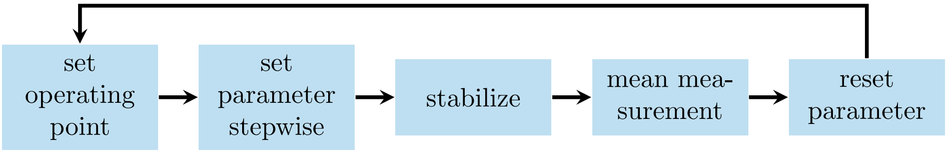


Figure 2.13: Common basic steady-state test bench measurement procedure

The average model quality is another main reason to utilize an adaptive test design procedure. Different to an online optimization, the search for an optimal solution is executed in an offline, post-processing step. Therefore, the model has to reflect the engine behavior in the whole measurable domain as good as possible. A common method to judge the generalization capability of a data-driven model is to separate the measurement data into a training and a test data set [Har15; RRC19; Fah+13]. The deviation of the model prediction and the measured test data provide an information about the model quality. The average model quality is judged by the mean squared error (MSE) [Fah+13] or root mean squared error (RMSE), given the model prediction and test measurements.

The test duration is the third criterion that needs to be examined. It is mainly influenced by the test automation and the task. The automation procedure and its optimization is not part of this research. However, there is an automation procedure affecting criterion that has to be fulfilled by the test design. The time between a new measurement is provided by the automation system and a new test point has to be provided by the design methodology is very restricted. In case the calculation duration is too long, the test automation suffers from a delay. That is, the model quality per time ratio is another important criterion. Instead of directly incorporating the ratio, an assumption of a minimum time per measurement of 300 seconds is introduced. If the steady-state measurement procedure is examined, the available duration is shorter though. A typical automation system sequence is the one shown in figure 2.13. Most important is the duration between the end of the measurement and the action to set a next operating point. The test design algorithm's available duration to provide a next test point must not be longer. A faster reset of the set parameter compared to the approach is common, mostly because there will be no limit violation and the base point is drivable per definition. Sometimes a direct reset is executed as well. Hence, this duration is set as criterion to a maximum of 30 seconds from experience.

The introduced open questions are examined during the following chapters. Each chapter covers more than one question. The following chapter 3 mainly treats the questions *piecewise-defined limit behavior* and *a continuous measured input domain volume increase*, where the other questions are considered as constraints only. Chapter 4 deals with an answer for an *optimal adaptive test design strategy* for *more than one output* but does not consider boundaries. Chapter 5 combines the developed strategies to find an answer to all questions, given the test design calculation duration constraint.

3 Adaptive Test Space Restriction

One main criterion a *1-Stage Full Adaptive* test design procedure, as defined in chapter 2, has to fulfill is the adaptive boundary search. A boundary model must be selected or developed to design the boundaries for a test point selection. The model is the base to describe the search area for any type of test design algorithm. Steady state information about piecewise-defined engine limits like misfire is rarely provided by the measurement procedure, which is why a regression-based model shows a major drawback and is not considered. There are possibilities to incorporate this information, by training a dynamic model to the parameter adjustment path and perform a steady state estimation for example. This method yet contradicts the requirement of an easy to apply adaptive test design and is therefore not considered during this research.

A boundary model selection is mainly affected by the assumption of the boundary shape. Different model types are introduced in section 2.3.3, where most of the models design a non-convex boundary shape. Compared to a convex boundary modeling, all non-convex methods are substantially more complex regarding either mathematical operations, tuning factors, or both. The application of such a model type should only be done if the process guarantees to show non-convex boundary behavior. Otherwise a convex hull approach is a better option from tuning perspective because there is a unique parameterless solution. However, a convex hull also has some drawbacks. The calculation time and memory demand raises exponentially with the dimension. In high-dimensional input domains the hull is not calculable and storable with common computational equipment. If the convex hull is applied to an adaptive test design, an extension strategy needs to be developed as well. The convex hull of a given point set only designs the interior of this point cloud, but the feasible operating range always is larger.

This chapter is organized as follows. The first section 3.1 describes for which reason only convex boundaries are taken into consideration in this research. The subsequent section 3.2 studies the possibility to design a convex hull within high-dimensional input domains and suggests how to use this boundary model within an adaptive test design approach. In section 3.3, procedures are developed to expand the convex hull area, which is necessary for an application within an adaptive test design. The section 3.4 applies the designed algorithms to a simulation and compares the results to a non-adaptive approach before the chapter is summarized in the last section 3.5.

3.1 Definition of Boundary Shape

One often discussed topic in model-based calibration is the occurrence of non-convex boundary shapes. Many studies deal with modeling and comparison of boundary models regarding synthetic or designed non-convex shapes. However, only few test bed measurements are published that show non-convex boundaries with a test design planned in a convex area. In [Kie14], synthetic data is generated from real test bench measurements. The results show slightly non-convex behavior in case an artificial limit is applied to the standard deviation of mean indicated cylinder pressure within a test case with camshaft variation in the whole engine operating range at a gasoline engine. Unfortunately, the used limit values are not discussed. The tests highlight a different level of non-convexity for different input domain dimensions but with the same model, which indicates changed artificial limits between the tests.

To the author's best knowledge, there are only spark timing boundaries described in literature that show significant non-convex boundary shapes in terms of combustion engine calibration where a convex test plan is applied. Slightly non-convex shapes often occur due to the limit behavior of the automation system and the non ideal reproducibility of combustion engines. Additionally, the test definition could offer non-convex definitions of input limits. The following sections introduce methods to deal with spark timing boundaries and a non-convex test domain definition without losing information when a convex boundary model is used. The precision improvement of modeling slightly non-convex boundaries, caused by the test automation or reproducibility, are not as significant as the gain in mathematic simplification of the hull model and all applicable methods. Hence, a convex boundary model is the base for all developed methods, where a solution is given for the two non-convex cases.

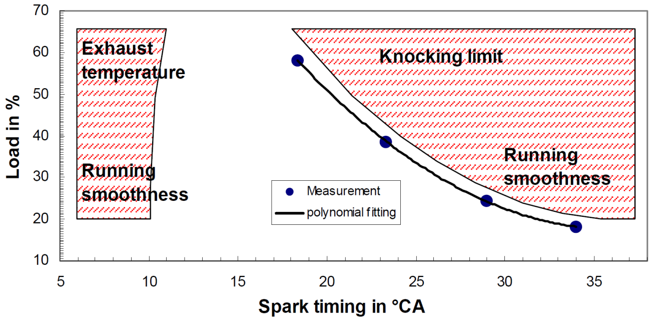


Figure 3.1: Non-convex engine operating boundary varying the spark timing and engine load [KSR06]

3.1.1 Spark Timing Boundaries

The concave hull determination method as shown in figure 2.11 (I) and introduced in [Kow17] is developed based on the knocking boundary behavior of gasoline engines. The base for this development is the measured, non-convex boundary shape in [KSR06] of engine knocking and running smoothness varying engine load and spark timing, as shown in figure 3.1. In most steady-state calibration tasks, the spark timing is no variation parameter for the engine model. A very common use case is to identify the engine behavior at the optimal combustion spark timing, which means a combustion center of 50% mass fraction burned (MFB50) at 8° crank angle (CA). The spark therefore needs to be controlled by the automation system during the adjustment of parameters, which easily can be done by a test bench automation system such as A&D ORION [AD20] with a parallel running controller for example. The controller also needs to react to knocking and maximum in cylinder peak pressure and operates the spark timing to stay near the boundary. Usual engine control units provide a functional correlation between spark timing and engine torque that has to be calibrated. Since the engine torque has nearly no delay regarding the spark timing, a very fast spark sweep from knocking border or maximum peak pressure to exhaust temperature limit or misfire can be executed as published in [Röp+07]. To derive the torque shape, there is no need of fitting a dynamic model. In case an exhaust temperature model or an emission model is required, a dynamic model with steady state estimation is applicable, where, however, the post processing effort is significantly higher.

Nevertheless, the non-convex boundary shape described by knocking and running smoothness is replaceable by a full spark sweep and an independent modeling. Since this procedure is state of the art, easy to apply and does not significantly increase the measurement duration, the spark timing boundaries are no reason to apply a complex non-convex boundary model to an adaptive test design.

3.1.2 Non-Convex Test Domain Definition

An often occurring non-convex input domain appears if constraints are applied by the calibration engineer. As discussed in section 2.3.4, different types of constraints are applicable to the hypercube input domain in the beginning of an MBC process. Especially map-based input constraints (see figure 2.12) lead to non-convex shapes per definition, except the base map is plane. Another typical task in both gasoline and diesel engine calibration is to find the optimal injection pattern. Defined distances between the end and the beginning of two sequential injections must be met for hardware reasons. Also the start of injections must be in sequence and has to be considered by the test design. All of these injection constraints are applicable by inequations that lead to a non-convex input domain.

In terms of an adaptive test design strategy with incorporation of engine boundaries, these constraints do not need to be considered by a boundary model. The test domain is restricted by the defined inequations and table- and map-based constraints. The boundary model only needs to represent additional engine limits occurred during the test. The resulting input domain is the intersection set of the boundary model including its extension, as will be shown in section 3.3, and the predefined input constraints.

3.2 High-Dimensional Boundary Model

The amount of convex hull-describing hyperplane elements rises exponentially with the dimension and in worst case also with the amount of test points. An example of number of hyperplanes for a varying amount of test points and with increasing dimension is shown in figure 3.2. The test points are randomly sampled on the surface of a unit sphere, which is the worst case scenario for a convex hull calculation, because all points are part of the hull surface. The calculation is done with the qhull algorithm

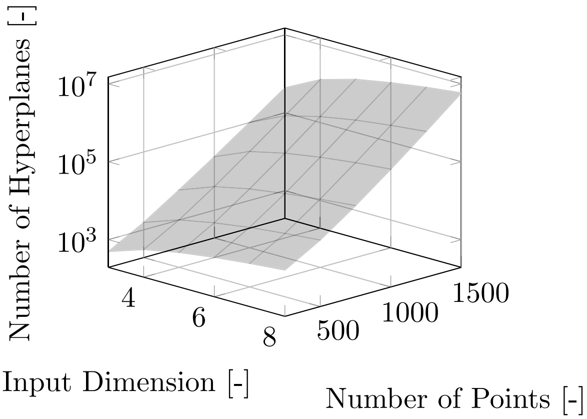


Figure 3.2: Amount of necessary hyperplanes to describe a unit sphere with raising dimension and amount of points

[Bar19], which automatically removes tiny hyperplanes, which is why the calculation is not deterministic but gives a good imagination. In case the hull described by hyperplanes has to be stored or used for an evaluation, a normal vector and a distance value has to be determined for each element. The normal vector size again grows linearly with the dimension. To get an idea about the computing duration: The calculation given 1500 points in 8 dimensions lasts approximately 167 seconds on an Intel® Core™ i5-7300U CPU with 2.6 GHz using the convhulln function in MathWorks® MATLAB® R2016b, which calls the qhull algorithm. The calculation of the normal vectors and distances is not included.

For adaptive test design methods with more than 10 inputs the determination of the convex hull is not suitable. Hence, a convex set algorithm is introduced that tries to find a solution for (2.63) given a test point and a hull-describing point set. The Gilbert-Johnson-Keerthi algorithm (GJK), presented in [GJK88], is an iterative procedure, which selectively tests hyperplanes for their orthogonal distance to the given test point. The main difference to a convex hull procedure is the advantage that not all hull-describing hyperplanes need to be calculated to check a given point. A detailed description of the algorithm can be found in appendix A.

Depending on the use case the one or the other method is faster. Once the hull hyperplanes are determined, the duration to evaluate a test point is faster than applying the GJK algorithm. As long as the hull-describing

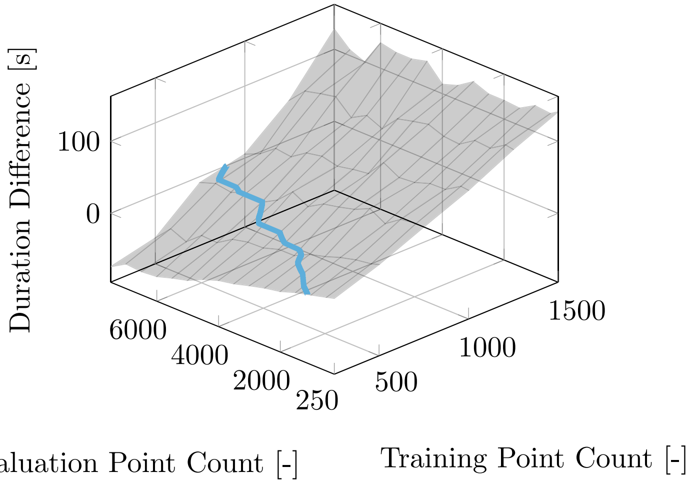


Figure 3.3: Calculation duration difference between convex set evaluation with GJK algorithm and convex hull evaluation including hull training for randomly sampled training and evaluation points in an 8-dimensional domain. Positive values state a benefit of convex set evaluation duration. The zero crossing shows the equality in evaluation duration for both methods.

dataset is used for the evaluation of a high amount of test points, the convex hull hyperplane calculation overhead is legitimated by the shorter evaluation duration. In an adaptive test design procedure, the convex hull could change with every measurement and the amount of test points to check is comparatively low. An example for the trade-off between both methods given the amount of test points and amount of hull training points is shown in figure 3.3 for an 8-dimensional case with randomly created test and hull training points. The duration difference contains both the hull training and evaluation duration for the convex hull determination method. The zero crossing line indicates where the transition from one to the other method is suitable. Increasing the amount of training points the advantage of the GJK algorithm raises. Also the benefit of convex hull evaluation for a higher amount of test points is lowered. With lower dimension the zero crossing takes place at a higher amount of training and evaluation points, which highlights that the convex hull determination method rather should be applied at lower dimensions. This information leads to the conclusion that the convex hull only should be used with a low amount of training points and at low dimensions. The zero crossings for

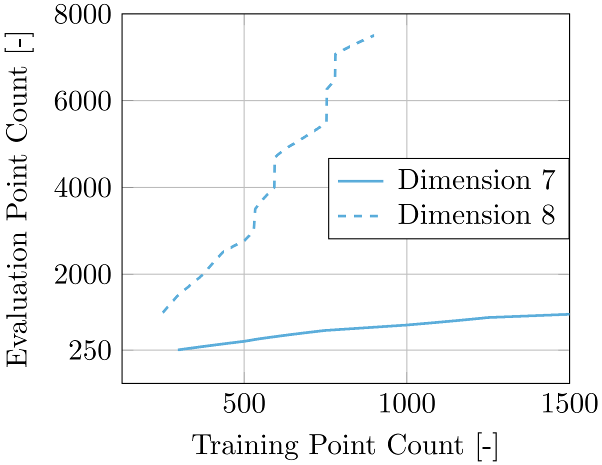


Figure 3.4: Zero crossing (see figure 3.3) for different dimensions for randomly sampled points. The area above each line shows a faster calculation for the convex hull approach. The area beneath indicates a faster calculation if the GJK algorithm is used.

different dimensions more accurately suggest where which algorithm should be used. In figure 3.4 the zero crossing lines are shown in relation to the amount of training and evaluation points for dimension 7 and 8. With this scope of point amount, no zero crossing exists at lower and higher dimensions. At dimensions less than 7, the convex hull approach is faster and at dimensions higher than 8 the GJK algorithm performs better.

Within an adaptive test design method, it is important how the boundary constraint is applied. The main two different ways of a next test point search are a global or local optimization of the cost function with constraints or an evaluation of the cost function with only candidates fulfilling the constraints. During an iterative optimization, no matter if it is a global or local optimization, the boundary constraint needs to be evaluated at each step. This type is applicable if the convex hull equations are calculated and the evaluation effort is low. In case the GJK algorithm is used, the function calls need to be as low as possible. This behavior should be kept in mind when introducing a test point search strategy.

The search area for a next test point must not be the convex hull area only. In case this region is applied only, a further determination of the true boundaries at a given stage is not possible, assuming a strict convex shape.

Therefore, the search area has to be expanded as to define a new region for a test point selection where measurements are likely feasible. A method is introduced in the following section to extend the convex hull boundaries.

3.3 Search Area Expansion

A search area should be defined by a combination of existing knowledge about engine boundaries, regardless of whether they are engine damaging or calibration target limits, and a boundary shape estimation in the unmeasured area. The most adaptive test design procedures presented in section 2.4 utilize an output-driven model to estimate the boundary shape, which has the main disadvantage to not consider piecewise-defined limits. A good alternative solution is the "potato model" that extends the range between hull-describing points. A disadvantage of this solution is the necessity for a tuning parameter. The calibration engineer is forced to apply a radius to define how fast the hull extends between boundary points, which is hard to guess, especially for different dimensional problems and tasks.

A solution for the search area definition is the linear extension of convex hull hyperplanes, as figured out in [SS17a], where the extrapolated area prohibits test points that do not strictly keep the convex hull definition. The idea is to define a sub-area driven by each boundary point and defined by the particular joint hyperplanes. Only as boundary labeled hull points are considered because those give rise about the true boundary shape. Instead of extending the convex hull, a next test point simply must not lie within any of the sub-areas. The definition of the exclusion area B is exemplarily shown in figure 3.5 for a 2-dimensional input domain with no other constraints and three boundary labeled hull points. Depending on whether a convex hull with all hyperplane definitions is known or not, the method to identify the exclusion area B is a different one. The next two sections introduce procedures for both conditions.

3.3.1 Hyperplane-Based Exclusion Procedure

In case the convex hull with all its hyperplanes, given by normal vectors and origin distances, is known, B can be identified by using the joint boundary point hyperplanes to evaluate the Hessian normal form (2.62). A candidate set $\boldsymbol{S}$ can be checked for being part of B or not by a sequential procedure.

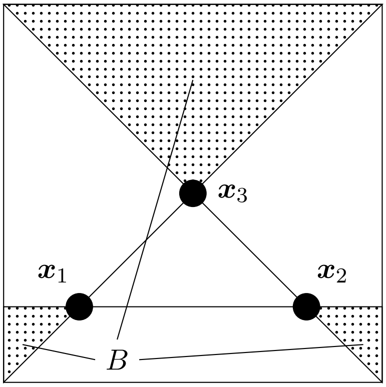


Figure 3.5: Definition of the exclusion area with predicted sub-areas B and labeled boundary points $\boldsymbol{x_i}$

Each as boundary labeled hull point is considered and the joint hyperplanes with normal vector and origin distance are selected. The distance d of a candidate $\boldsymbol{s}_k$ is calculated by the Hessian normal form, where $\boldsymbol{s}_k$ is part of the sub-area in case all distances are positive. This contradicts (2.62), because the normal vector of a convex hull points outside the hull and therefore a point is part of the sub-area if it lies on the other side of all joint hyperplanes. Figure 3.6 points out this behavior. The algorithm stops early in case all candidates are already assigned to any sub-area. In case a low-discrepancy candidate set is provided the early stop is very unlikely. In a sequential procedure with only a single candidate check this abortion criterion is necessary to avoid unnecessary calculations. The full algorithm to test a candidate set for being part of B or not is shown in algorithm C.1 in appendix C.

During engine test bench measurements the applied test point distance to the boundary can differ due to a changing automation system configuration for example. Even though it should be avoided, in practice it happens that the parameterization is changed by the calibration engineer. Therefore, slightly non-convex boundary measurements can occur, which result in an error, because the boundary labeled point is not part of the convex hull. Those points need to be projected onto the convex hull surface and integrated to it. A simple procedure is applied as follows. The hyperplane with lowest distance to the inside lying boundary point is determined by

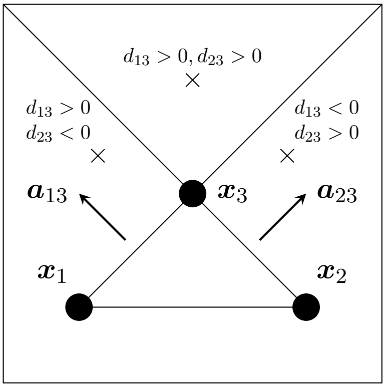


Figure 3.6: Hyperplane normal vector and distance definition for the joint hyperplanes of boundary point $\boldsymbol{x}_3$ with normal vectors $\boldsymbol{a}_{13}$, $\boldsymbol{a}_{23}$ and resulting distances d.

the Hessian normal form distance calculation. The intersection of the normal vector of this hyperplane, starting at the boundary point, and the hyperplane is identified solving the linear equation

$$\boldsymbol{x}_j + \lambda \boldsymbol{a} = \boldsymbol{o} + \boldsymbol{\delta D} \tag{3.1}$$

where $\boldsymbol{o}$ is any boundary point of the considered simplex and $\boldsymbol{D}$ is a matrix of simplex direction vectors, preferably calculated by all other joint boundary points of the simplex. The result for the intersection point can be derived by solving

$$\begin{bmatrix} \lambda \\ \delta_1 \\ \vdots \\ \delta_{n-1} \end{bmatrix} = \left(\begin{bmatrix} a_1 & d_{1,1} & \dots & d_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_n & d_{n,1} & \dots & d_{n,n-1} \end{bmatrix}^\top \right)^{-1} \begin{bmatrix} o_1 \\ \vdots \\ o_n \end{bmatrix} \tag{3.2}$$

and applying either λ or $\boldsymbol{\delta}$. The determined intersection point is added to the hull and incorporated during the exclusion area check.

3.3.2 Iterative Exclusion Procedure

The iterative exclusion procedure takes place if the hyperplanes of a set of points are not determined, i.e. when the GJK algorithm is applied instead of a convex hull calculation. Since the joint hyperplanes of a labeled boundary point are not present, a different calculation has to be applied to test a given candidate about its affiliation.

Basically, the joint hyperplanes of a convex hull boundary point equal the convex cone generating hyperplanes of the point set and can be inferred by the convex cone and convex set definition. A convex cone is defined by

$$\mathrm{cone}(\boldsymbol{X}) = \left\{ \sum_{i=1}^{k} \alpha_i \boldsymbol{x}_i \,\middle|\, \alpha_i \geq 0 \;\forall i \right\} \tag{3.3}$$

which is the same representation of an area as (2.63), except that the condition about the α_i is different. The missing sum condition allows to quit the convex hull area and also includes the origin to the convex cone. However, a negative α_i is prohibited, which is why the joint hyperplanes of a boundary point, if given as the origin, are exactly the convex cone limiting hyperplanes. To find a linear combination for a given candidate $\boldsymbol{s}_k$ for (3.3) there is no closed solution. An iterative procedure, derived from the GJK algorithm [GJK88], is presented in [ZC09] to find the closest point on cone($\boldsymbol{X}$) for $\boldsymbol{s}_k$. A detailed explanation of the algorithm is given in appendix B.

The coordinate system origin is the same as the convex cone origin if the cone is defined by (3.3). Hence, the cone-generating points and the candidate at first need to be translated by the considered boundary point $\boldsymbol{x}_i$ by $\boldsymbol{x}_j - \boldsymbol{x}_i \,\forall\, \boldsymbol{x}_j \in \boldsymbol{X}$ resulting in $\boldsymbol{X}_t$ and by $\boldsymbol{s}_t = \boldsymbol{s} - \boldsymbol{x}_i$. Similar to the hyperplane-based method, the cone-generating hyperplanes need to be negated. Another, more appropriate way is to mirror the candidate at the new origin and test for being part of the convex cone or not. That is, if and only if $-\boldsymbol{s}_t \in \mathrm{cone}(\boldsymbol{X}_t)$, then $\boldsymbol{s}_k$ is part of the sub-area. This check has to be done for all as boundary labeled points to determine if $\boldsymbol{s}_k \in \boldsymbol{B}$. The full algorithm to check a candidate set $\boldsymbol{S}$ for being part of the exclusion area B is shown in algorithm C.2 in appendix C.

A characteristic of the convex cone algorithm is, that in case the boundary point under consideration is slightly inside the convex hull the whole input domain is identified as inside the convex cone. During engine test bench measurements the applied test point distance to the boundary can differ depending on the automation system configuration for example. Therefore, slightly non-convex boundary measurements can occur, which force the convex cone algorithm to exclude all candidates in the input domain.

All labeled boundary points need to be checked if they belong to the convex hull building points, where the GJK algorithm can be applied for. Labeled boundary points that do not contribute to the convex hull first have to be projected onto the hull surface. This is quite important for a simple reason. In case points are part of the same hyperplane, where the non-hull contributing point is labeled as boundary, a huge area of the non-measured area can be defined as exclusion area. Figure 3.7 illustrates this behavior. The hyperplanes are not known, which is why a simple orthogonal projection onto the surface, as shown in the hyperplane-based strategy, is not possible. Another procedure is introduced to find an almost optimal solution. The aim is to only project the considered boundary point onto the most appropriate hyperplane. The criterion, which is used, is a support vector starting at the hull center point to the considered point. The hull center point $\boldsymbol{\gamma}$ is calculated as the center of gravity of all points contributing to the hull, which is, for simplicity of calculation, not the volumetric hull center. To determine the volumetric hull center, a Delaunay triangulation of the boundary points would be necessary, which calculation duration contradicts the gained benefit of a convex set application. Hence, the support vector is calculated and normed to the unit length by

$$\boldsymbol{v}_{sn} = \frac{\boldsymbol{x}_j - \gamma}{|\boldsymbol{x}_j - \gamma|}. \tag{3.4}$$

For all points $\boldsymbol{x}_i \in \boldsymbol{X}$, a support value $\rho(\boldsymbol{x}_i)$ is calculated by

$$\rho(\boldsymbol{x}_i) = (\boldsymbol{x}_i - \boldsymbol{x}_j)\boldsymbol{v}_{sn}^{\top} \tag{3.5}$$

considering the as boundary labeled point $\boldsymbol{x}_j$. The point $\boldsymbol{x}_i$ with the highest support, which is the point that has the furthest projection on the support vector, is projected onto the support vector by

$$\boldsymbol{x}_{ip} = \rho(\boldsymbol{x}_i)\boldsymbol{v}_{sn}. \tag{3.6}$$

The point $\boldsymbol{x}_{ip}$ is located on the convex hull hyperplane that intersects the extended vector $\boldsymbol{x}_j - \boldsymbol{\gamma}$ if the intersection is perpendicular or slightly outside the hull. Applying the GJK algorithm to $\boldsymbol{x}_{ip}$ to find the closest point on the convex hull delivers a surrogate point $\boldsymbol{x}_{ih}$ on the intersected convex hull hyperplane.

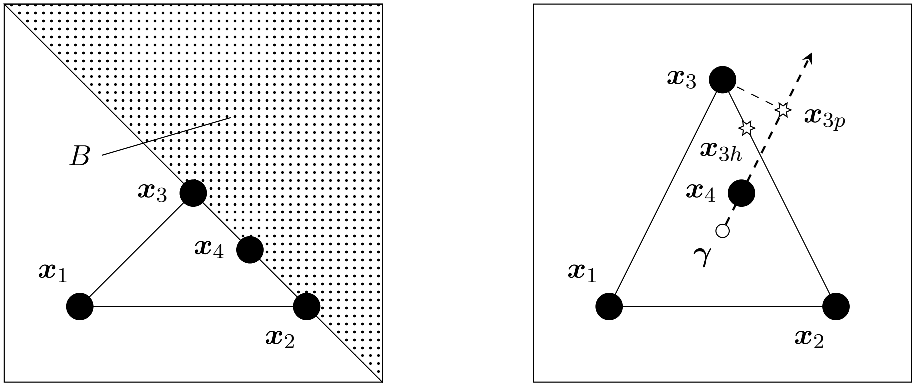


Figure 3.7: Left: Exclusion area definition in case 3 points are part of the same hyperplane. $\boldsymbol{x}_4$ is labeled as boundary point and therefore the joint hyperplanes hold the same normal vector and origin distance. Due to the convex criterion, a large area is defined as exclusion area B. Right: Construction of a surrogate point for an in hull located boundary point. A support vector is calculated by the hull point center of gravity $\boldsymbol{\gamma}$ and the boundary point $\boldsymbol{x}_4$. The furthest support point $\boldsymbol{x}_3$ is projected onto the support vector to $\boldsymbol{x}_{3p}$. The nearest hull point $\boldsymbol{x}_{3h}$ for $\boldsymbol{x}_{3p}$ is determined by the GJK algorithm.

In the very rare case the furthest boundary point lies on the support vector, the surrogate point $\boldsymbol{x}_{ih}$ is the same as the boundary point, which is not the target of the algorithm and a wrong solution. In this case, a random noise is added to the support vector $\boldsymbol{v}_s$ and it is normed again to change the support vector direction slightly. The full algorithm is shown in algorithm C.3 in appendix C.

The iterative exclusion procedure is called iterative, contrary to the hyperplane-based method, because a candidate set has to be checked iteratively and there is no joint calculation for several candidates. The distance determination within the hyperplane method for one boundary point is a matrix operation, which is jointly calculable for a whole candidate set. However, both methods have to loop over all labeled boundary points or until the abortion criterion is reached. The iterative procedure suffers from a higher amount of operations compared to the hyperplane-based method. Depending on the complexity of the test design algorithm, it is more reasonable to first calculate a next test point and then check this candidate for being part of the exclusion area or not. The erased candidates are persisted and not accounted in a subsequent test point search, because the convex assumption does not allow them being inside the measurable area again.

3.4 Simulation-Based Investigation

A verification of the theoretical considerations is done regarding the requirements listed in section 2.5. A comparison measurement at an engine test bench is very expensive in terms of both money and effort, which is why the verification is done by simulation models. This additionally offers the possibility to run repeated tests for statistical validity. The subsequent sections define the objectives in detail, the architecture of the simulation model, and show the results of a comparison to a non-adaptive approach.

3.4.1 Objectives

The two main objectives to verify are the continuous increase of the measured input domain size and the increase of the average model quality. The measured input domain size is calculated by means of the convex hull calculation and the volume the hull incorporates. The MATLAB® function convhulln is called in each iteration to determine the hull volume. A steady increase and a high volume result are the main criteria.

The average model quality is determined by a huge validation set. A Sobol sequence is generated in the whole input domain. Points are iteratively added to the validation set, which fulfill all boundary constraints of the simulation model. 10000 validation points are generated by this method and evaluated by the simulation model in each iteration without adding any noise. The validation RMSE r is calculated as the RMSE of the noise free validation points and the model predictions.

A model structure independent test design is chosen to focus the effect of the boundary detection method only. Hence, new test points are generated by the maximin algorithm, given a Sobol sequence with 30000 candidates, which is different to the validation point sequence. The boundary incorporation-based method is compared to a strict space filling, planned in advance into the unit hypercube input domain $[0,1]^{\text{dim}}$. The first 10 test points are designed similar for both methods. For the adaptive approach an iterative test design is applied in each iteration. A next test point is planned by maximin and the result is checked for being part of B. In case

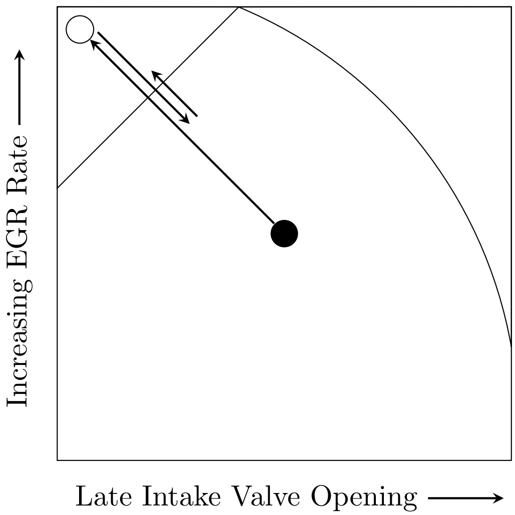


Figure 3.8: Bisection method with 2 iterations to find a near boundary surrogate measurement. The outer area marking is not shown for better visualization of bisection, only the boundary shape is shown. Example from figure 2.7.

it is a part, the candidate is erased from the candidate set and a next test point is planned by maximin until a solution is found that is not part of B.

If a planned test point is not within the simulated boundaries, a simple iterative bisection method takes place to find the closest surrogate measurement location within the boundaries. For this surrogate measurement identification, a vectorial adjustment is assumed, starting at a given base point. The bisection method is exemplary shown in figure 3.8 for two iterations. During the simulation the bisection is iteratively applied until the bisection length reaches a threshold of $\epsilon_t = 2.22 \cdot 10^{-11}$ and the solution is within the bounds. The simulation model is executed at this surrogate location and the output is used for model training. Both, the original test point and the surrogate location are used as inclusions for the subsequent adaptive test design iteration.

3.4.2 Simulation Model

The impact of design space restrictions on the model quality is different for various processes. Within this simulation, the relevant factor for the model quality is the space-filling test design quality that is affected by the design space restriction. The identification of a strong oscillating or nonlinear process, however, is much more influenced by a low space-filling quality than the identification of a linear one. Different simulation models are introduced to demonstrate the quality change provoked by the boundary incorporation strategy. In engine test bench measurements, the same input setting does not exactly reproduce the same output if measured several times. Different to computer experiments, the measurements exhibit diverse noise types. A simulation of engine test bench measurements therefore has to incorporate artificial noise. Both, the simulated processes and the artificial noise model are introduced in the following sections.

Process Models

The goal of the developed methods is to improve the identification of measurable variables to describe the combustion engine behavior by means of steady state test bench measurements. Dependent on the type of engine, the detailed engine hardware, and the calibration target several variables are of interest that behave different and sometimes show opposite trends on an input variation. The most relevant engine calibration parameters have smooth gradient changes. However, some are more difficult to model like for example soot emissions in both diesel and gasoline engines. Four different mathematical model types are implemented and used for the simulative verification.

A nonlinear function, called radcosn, is provided by [Pol02] and aims in a two dimensional case (radcos2) at representing the behavior of the fuel consumption of gasoline engines while varying the intake and exhaust camshaft [Rai15]. This function is referenced by several optimization and test design algorithms (see e.g. [Zag14; Rai15]) because it represents a realistic problem with several local minima. The generic, multidimensional radcosn function is described by

$$
\begin{aligned}
y_{radcosn}(\boldsymbol{x}) = & \cos\left(9\sqrt{x_1^2 + x_2^2} + 2\right) \\
& + \frac{1}{2}\cos(8x_1(1 + x_3 + ... + x_{\text{dim}}) + 2) \\
& + 15((x_1 - 0.4)^2 + (x_2 - 0.4)^2)^2 \\
& + \frac{15}{\sqrt{\text{dim} - 2}} \sum_{i=3}^{\text{dim}} (x_i - a_i(b_i x_1 + 1 - x_2) - 0.2)^2
\end{aligned}
\tag{3.7}
$$

with $b_i = 1 + 5(i - 3)$, $a_i = \frac{0.6}{1+b_i}$ and dimension dim. The radcos2 function is exemplarily shown in the bottom left plot in figure 3.9.

Another smooth but strong nonlinear process is a mixture of a hyperbolic and a Gaussian function, introduced in [Har14] to compare different combustion engine modeling algorithms. The function is defined by

$$
y_{GH}(\boldsymbol{x}) = \frac{1}{1 + 10\sum_{j=1}^{\text{dim}} \frac{1}{\text{dim}}(1 - x_j)} + \frac{29}{44}\exp\left(-\frac{1}{2}\sum_{i=1}^{\text{dim}}\left(\frac{x_i}{0.25}\right)^2\right) \tag{3.8}
$$

for a given input vector $\boldsymbol{x}$ with dimension dim. This function also shows several local minima and high gradient changes but at the same time has a smooth behavior, which represents typical combustion engine characteristics. The mixture model is shown in a two dimensional case in figure 3.9 bottom right.

The CEC2009 competition [Zha+09] provides different models, where the cost function for an optimization shows several optima. For all models a Pareto front exists, which is the same especially in diesel engine optimization. The constrained problem 1 consists of two functions that have an inverse trend. Additionally, a constraint is defined to serve as a boundary information for the simulation environment. The first output is defined by

$$
y_{CEC1}(\boldsymbol{x}) = x_1 + \frac{2}{|\boldsymbol{J}_1|}\sum_{j \in \boldsymbol{J}_1}\left(x_j - x_1^{0.5\left(1 + \frac{3(j-2)}{\text{dim}-2}\right)}\right)^2 \tag{3.9}
$$

and the second by

$$y_{CEC2}(\boldsymbol{x}) = 1 - x_1 + \frac{2}{|\boldsymbol{J}_2|} \sum_{j \in \boldsymbol{J}_2} \left(x_j - x_1^{0.5\left(1+\frac{3(j-2)}{\dim-2}\right)} \right)^2 \tag{3.10}$$

with $\boldsymbol{J}_1 = \{j|j \text{ is odd and } 2 \leq j \leq \dim\}$ and $\boldsymbol{J}_2 = \{j|j \text{ is even and } 2 \leq j \leq \dim\}$. The search space constraint is given with

$$y_{CEC1} + y_{CEC2} - a|\sin\left[N\pi(y_{CEC1} - y_{CEC2} + 1)\right]| - 1 \geq 0 \tag{3.11}$$

with N being an integer and $a \geq \frac{1}{2N}$. The 2009 contest specifies $N = 10$ and $a = 1$, which is adopted for the simulation model. The functions are exemplarily shown in the two upper plots of figure 3.9 in a 3-dimensional input domain with fixed second input, because the first output function is defined for at least 3 inputs.

In addition to the artificial model types, a data-driven combustion engine model is applied as well. A diesel engine is measured with a space-filling test design at an engine test bench. Next to engine speed and engine load 7 further inputs are varied. The input impact on the relevant outputs

- carbon monoxide (CO),
- fuel mass flow ($\dot{m}_{\text{fuel}}$),
- exhaust temperature before turbine ($\text{T}_{\text{Bef Turbine}}$) and
- soot mass flow ($\dot{m}_{\text{Soot}}$)

is modeled by a GP model. The model configuration is shown in figure 3.10. The model is used with fixed speed and load values for simplification. With changing speed and load many different effects arise that make it more difficult to precisely judge the main effect of different test design algorithms. As an example, consider the adjustment of speed and load, which should not be done arbitrarily during a test run. Each operating point change takes a long time during an automated test run, which is why the test points are commonly sorted in a meandering shape. This problem is considered in chapter 5.

A restriction on the output of the artificial models needs to be introduced for the simulation of an artificial boundary. Even though the CEC2009

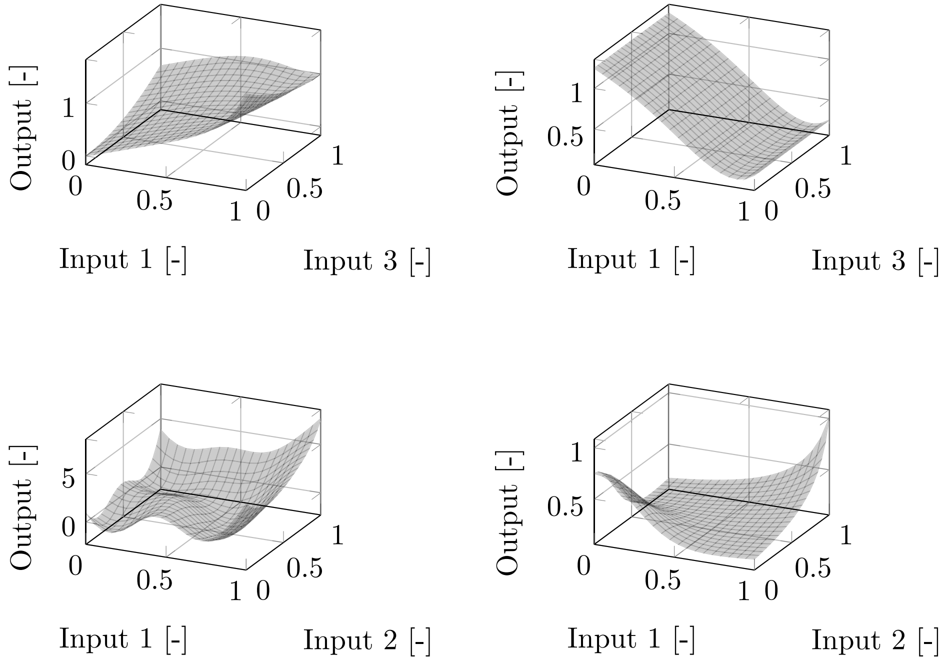


Figure 3.9: Representation of the used generic process models. Top left: First output of the 3-dimensional CEC2009 competition model with $N = 10$ and $a = 1$ with a variation of the first and third parameter [Zha+09]. Top right: Second output of the 3-dimensional CEC2009 competition model with $N = 10$ and $a = 1$ with a variation of the first and third parameter [Zha+09]. Bottom left: Radcos2 function [Pol02]. Bottom right: Mixture of hyperbolic and Gaussian function with two inputs [Har14].

competition model provides a constraint, more restrictive boundaries are introduced for all outputs additionally. Within the artificial model, lower bounds restrict the output and therefore the input domain. The outcome of the boundaries, especially the CEC2009 model boundary, is non-convex in some areas. A convex surrogate model is created, which is trained by 10000 Sobol-based candidates. Only those candidates fulfilling all constraints are the input to a convex set-based hull model. During the simulation, this hull model specifies the model boundaries. The diesel engine simulation model provides a convex hull model that corresponds to the real measured input domain and is used to define the valid evaluation range of the GP models within this simulation. No more constraints need to be applied for

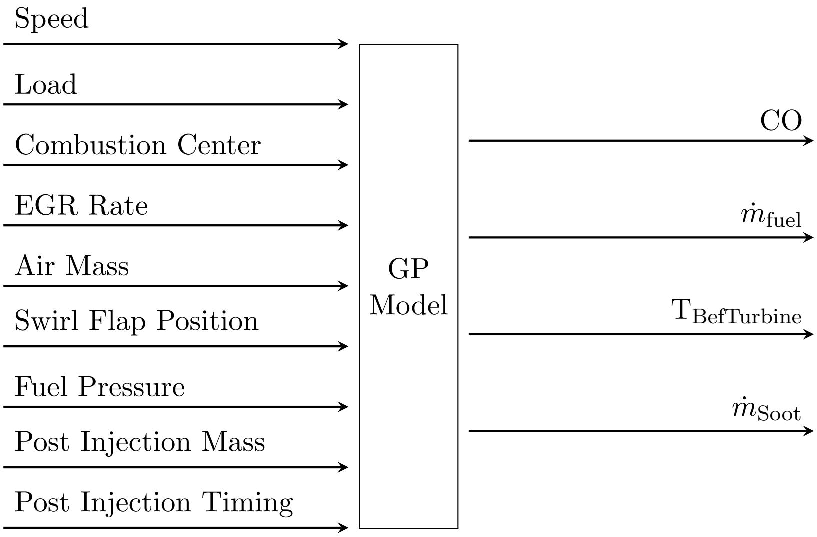


Figure 3.10: Diesel engine simulation model structure

this model type. An overview over the restrictions for both model types is given in table 3.1.

The diesel engine simulation model is always used with all inputs except speed and load. A higher input dimension is represented and tested by the artificial simulation models. These are calculable with an arbitrary amount of inputs, for which reason an input dimension of 9 is chosen to be a good compromise between a high-dimensional test and a common application of combustion engine identification.

Table 3.1: Boundary constraints definition

Model Type	Dimension	Lower Bound	Upper Bound	Further Restriction
Artificial	9	0.5, 0.4, 2, 0.15	-	CEC2009 Constraints
Diesel Engine	7	-	-	Convex Hull

Noise Model

The observation of engine test bench measurements does not only highlight simple i.i.d. Gaussian noise in the output signals. To approximate an output noise free steady-state value, a recording of all parameters is executed at the engine test bench for 30-60 seconds. Each focused parameter is averaged over the recorded duration to subtract parts of the measurement device noise. If only i.i.d. noise on the output would be present, a mean steady state measurement would be able to identify the real process output with high certainty. Unfortunately, the three types of noise

- actuator noise ϵ_i,
- plant noise ϵ_p and
- measurement device noise ϵ_o

are present. An oscillation in actuators is always present, because the ECU continuously controls the desired engine operating status. The oscillation amplitude and frequency is significantly changing for different ECU calibration stages and operating conditions. A part of the oscillation is considered by the mean measurement, but in case the input/output mapping is nonlinear the measurement deviates from the real process output for the averaged input setting. A second, hard to reproduce noise type is caused by the fact that a combustion engine behaves different over run time due to wear and carbonization. Also operating conditions such as the ambient pressure and humidity are typically not controlled in an engine test cell and have an influence on the engine behavior. These long-term effects also have to be considered by the simulation noise model as plant noise. The third noise type is the measurement device noise. The noise level is significantly reduced by the mean measurement but a drift of the measurement device occurs as well. Especially exhaust gas analyzers exhibit a short-term drift. A common procedure is to calibrate the exhaust gas analyzers daily to reduce the measurement device error.

The measurement device noise is assumed to be i.i.d. and is therefore considered by a Gaussian noise. The highest noise is expected for exhaust gas analyzers. Since these are calibrated daily, a good knowledge about their measurement deviation is present. A relative deviation of the calibration gas more than 1.5 % before calibration and 1 % after calibration is commonly accepted and is exceeded seldom. Therefore, the measurement device noise is assumed with a standard deviation of $\sigma_o = 0.01$, because in most cases 1 % deviation is not exceeded but there also exists a significant probability (31.73 % for this definition) that the error level is exceeded. The application of a plant noise is more challenging though. A measurement of the plant noise is very time consuming and expensive due to the long term effect. As a random trend noise a Wiener process is applied, because this type of noise shows a correlated noise given a duration. The Wiener process can be defined by the normal distributed change over a time step dt by

$$dW \sim \sqrt{dt}\mathcal{N}(0, \sigma_w^2). \tag{3.12}$$

A noise is added to each simulation model output by the definition of an independent Wiener process for each output. The time delta is assumed to be $dt = 1$ as a measurement-based time step, because no measurement is present to identify the real process noise. Each process sample deals as plant noise level for a single simulation. The noise magnitude is thus only controlled by σ_w^2. A systematic error has a variable impact on different parameters, which is why a noise value is generated for each model output separately but with the same noise variance. As an example the combustion chamber carbonization in gasoline engines can be taken into consideration. The knocking sensitivity is raised because the self ignition risk at hot carbon spots is higher than at the clean piston and valve surfaces. The gas exchange of fresh and exhaust gas is not influenced remarkably by the carbonization, while the exhaust emissions are influenced variously. A test design method has to handle these effects and therefore the simulation environment needs to provide the necessary conditions. From test point to test point, the trend and noise level can turn. The process noise indeed is not independent, but the dependency is hard to identify. The Wiener process noise is added to indicate a dependent noise randomly by its definition. Random samples from a Wiener process with $\sigma_w = 0.0005$ are shown in figure 3.11 and are found to be suitable for the reconstruction of real measurements. Both, the measurement device and plant noise are added as relative noise.

Next to the plant and measurement device noise the input noise is considered. A Gaussian i.i.d. noise represents the controller deviation from the demanded values during a measurement. Different than the output noise, the input noise and its influence on the output by the plant model is calculated 600 times for each mean measurement. This represents a typical measurement duration of 60 seconds with a sample rate of 10 Hz. The

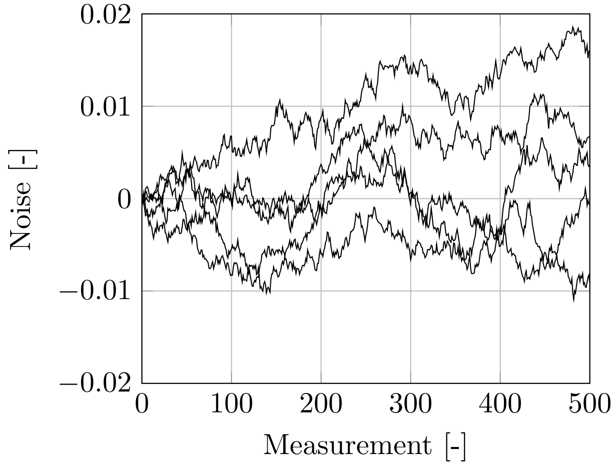


Figure 3.11: Wiener process samples over 500 measurements given a Gaussian standard deviation of $\sigma_w = 0.0005$ and a sample discretization of $dt = 1$.

variance of the input noise is assumed to be the same for each input and is added relatively as well. Investigations offer a linear noise magnitude dependency on signal magnitude in most cases. Even the MFB50 control noise has a strong dependency on the absolute value. It could be assumed to be independent on the output magnitude, because an MFB50 around the top dead center (0 °CA) offers noise as well. However, the dissenting behavior is explained by the more fluctuating combustion process at late MFB50 locations, which is reflected by higher controller noise. The simulation function with added noise is defined by

$$\begin{aligned} y(\boldsymbol{x}, j) &= \frac{\sum_{k=1}^{m} f(\boldsymbol{x}(1+\epsilon_i))}{m}(1+\epsilon_p)(1+\epsilon_o) \\ &= \frac{\sum_{k=1}^{m} f(\boldsymbol{x}(1+\sigma_i^2 \boldsymbol{\xi}_i))}{m}(1+W(j))(1+\sigma_o^2 \boldsymbol{\xi}_o) \end{aligned} \tag{3.13}$$

with ξ being a normal random variable with $\mathrm{E}(\xi) = 0$ and $\mathrm{var}(\xi) = 1$ and j as the measurement number. During the simulation, the function output with input noise is sampled 600 times for each output, meaning $m = 600$. The process and output noise is only added once.

Investigations of the input noise level in diesel and gasoline engines show a typical maximal averaged input noise standard deviation of $\sigma_i = 0.02$,

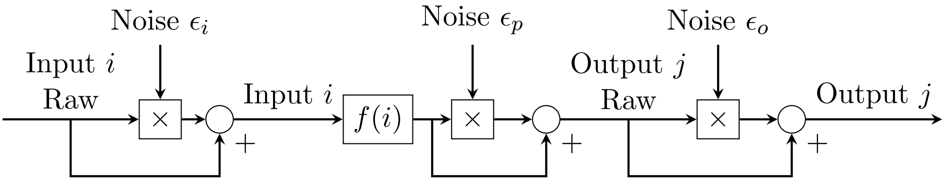


Figure 3.12: Overall simulation model architecture with i.i.d. Gaussian input noise ϵ_i, i.i.d. Gaussian output noise ϵ_o and Wiener process-based plant noise ϵ_p

which is related to the signal magnitude. Controlled physical parameters as for example the EGR rate, the air mass or the rail pressure exhibit a higher input noise than positioning controller as for example the variable valve timing or the injection timing. However, during the simulation a worst case scenario is created, where each input experiences an individual noise but with the same relative noise level. The overall simulation model including input, plant, and output noise is given in figure 3.12.

3.4.3 Discussion

Each simulation is run 20 times to judge the average improvement and fluctuation range. Figure 3.13 shows the mean relative validation error for each output for the diesel engine simulation model. The validation error r is normed to the sum of the arithmetic mean and the standard deviation of all validation point output values $\boldsymbol{y}_{val}$ for the particular output i by

$$r_{rel,i} = \frac{r_i}{\overline{\boldsymbol{y}_{val,i}} + \mathrm{std}(\boldsymbol{y}_{val,i})} \cdot 100. \tag{3.14}$$

The shadowed area is the standard deviation range, calculated by all 20 simulation runs at each simulation step. The validation error decreases, as expected, asymptotically with the number of training points. The fluctuation between the simulation runs seems to be different for the outputs but is a result of a different error level and is thus explained by axes scaling. The input variation has a strong influence on the CO and soot emissions but a lower one on exhaust temperature and fuel consumption. The simulated condition is only one operating point by what the low impact on exhaust temperature and fuel consumption is explained. To describe it in a different way: The CO and soot functions are considerably more complex in this test case compared to the fuel consumption and exhaust tempera-

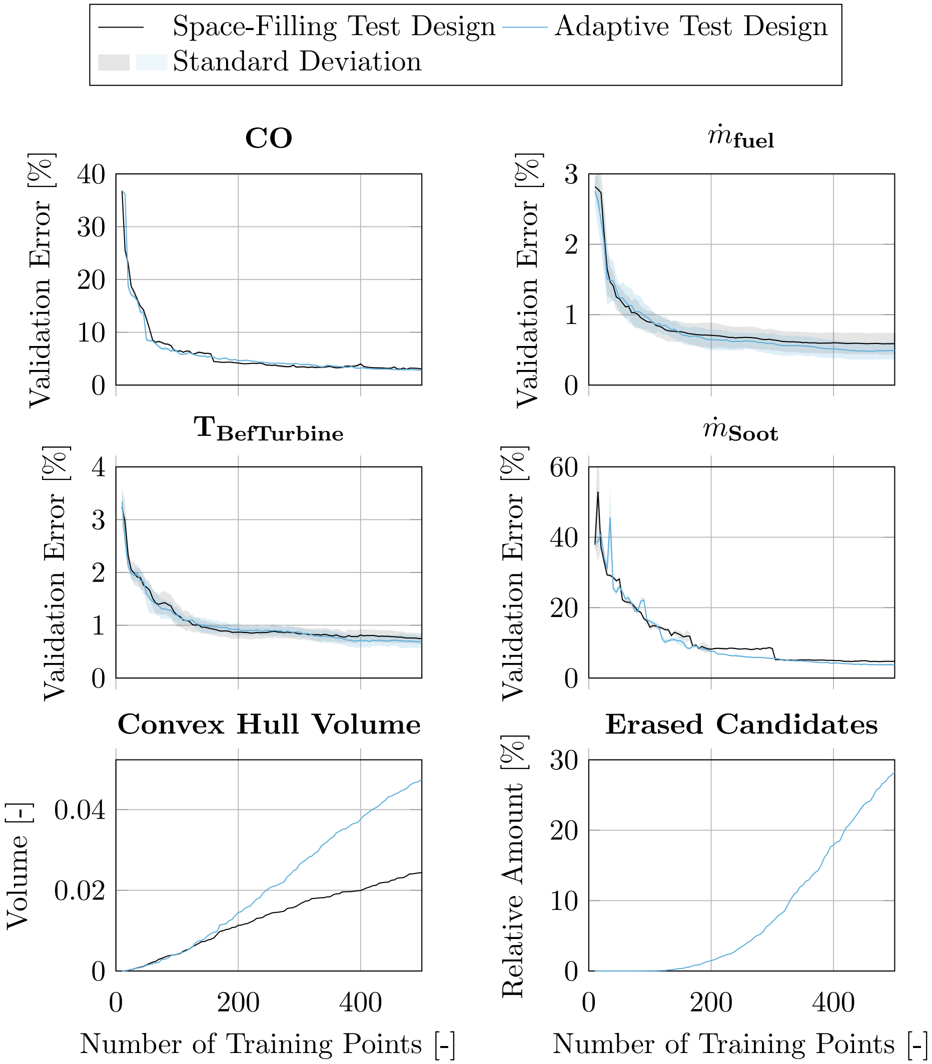


Figure 3.13: Top four plots: Relative validation error trend for all 4 outputs of the diesel engine simulation model with 7 inputs (model structure as defined in figure 3.10). The average validation RMSE is shown solid and the standard deviation between all 20 simulation runs is given as shadowed area for error stability estimation. Bottom left: Convex hull volume trend for the space-filling test design and the adaptive test design during the simulation of the diesel engine identification with 7 inputs. Bottom right: Relative amount of cumulated erased candidates for each iteration by using the adaptive test design method.

ture functions. This property can be recognized by the amount of error and therefore the different scaling. While the relative error of fuel consumption and exhaust temperature is generally very low, the error of the emissions is significantly higher. With both the adaptive and simple space-filling test design the models are able to learn the process behavior. The uncertainty in process identification is approximately similar for both methods. However, the adaptive test design is able to reduce the validation error slightly faster than the non-adaptive approach in most cases. The error decrease for the CO emissions is very similar but all other 3 outputs show a faster decrease of error for the adaptive approach. Also the validation error at the end of the simulation is lower in all cases for the adaptive approach.

Additional to the better process identification, the measurement covered input region is increasing significantly faster for the adaptive approach, as can be seen in figure 3.13 in the bottom left plot. Starting from 130 training points, a faster growth of the convex hull volume is achieved, resulting in a volume nearly twice as with the non-adaptive approach. The volume shown is normed to the unit hypercube to have a maximum value of 1. Since convex boundaries are present within the input domain, a volume of 1 is not reachable. The faster volume increase is explained by two different effects. Firstly the adaptive approach includes both the planned and the measured points to design the next test point. In case a test point is not applicable and a surrogate measurement is executed at the boundary, two inclusions are generated instead of only the designed test point as with the non-adaptive approach. The second and more important influence is the candidate erase strategy as introduced in this chapter. Once information about the boundaries is gathered the strategy applies and the exclusion area B starts growing. Boundary measurements near already taken measurements are not likely taken anymore. The bottom right plot in figure 3.13 shows this fact in dependency on the number of measured training points. The relative amount of candidates that are erased because they are part of the exclusion area B is shown. Simultaneous to the start of the exponential increase of erased candidates, the convex hull volume starts to increase significantly faster than with the non-adaptive approach. During an optimization, this considerably higher model evaluation range could lead to a better optimum that potentially would not be identified with the non-adaptive approach. Together with the lower validation error an optimal solution is also identifiable with higher certainty. Another view is offered by the evaluation of measurement duration decrease by means of the adaptive approach. The hull volume level with 500 non-adaptively planned measurements is reached with the adaptive approach by 290 measurements. This offers a measurement time reduction of 42 % by using the adaptive approach considering the measured input domain.

The simulation results of the artificial models in a 9-dimensional input domain are given in figure 3.14. A distinct validation error decrease trend by using the adaptive approach is not present for the CEC2009 competition model outputs. A slightly better, though not clearly differentiable decrease in the validation error of output 2 of the CEC2009 model is noticeable compared to the space-filling test design. The ambiguous behavior is indicated by the standard deviation as well, since it shows a high fluctuation between each simulation run, especially for the adaptive test design. A different situation is given for the radcosn validation error, which reaches a significantly lower value by using the adaptive test design. The test design is crucial for this function, because it exhibits several local optima and has a high dynamic shape. A severe resulting validation error level for both strategies points this out with a clearly higher validation error compared to all other outputs. At the Gaussian hyperbolic function the same lowest, saturated error is achieved for both test design methods, where the low error is reached significantly quicker with the adaptive test design. The overall validation error situation gives rise that the effect of the boundary estimation has a positive influence on the test design. No increase in validation error is determined and the validation error trend and resulting level is clearly improved for the radcosn and Gaussian hyperbolic function even in this comparatively high-dimensional input domain.

The resulting convex hull volume is significantly improved during the whole simulation as shown in the bottom left plot in figure 3.14. As with the 7-dimensional example, the hull volume starts to increase faster with the adaptive test design at that point where the exclusion area strategy starts excluding candidates. Candidates are erased continuously, while new boundary information is provided. After 800 measurements, a hull with approximately 45 % higher volume is the result of the adaptive test design strategy. Or from the efficiency point of view, the hull volume by 800 measurements with the non-adaptive approach is reached already with 585 measurements with the adaptive approach. This is a decrease of almost 27 % measurement duration. Even though the input domain offers a very

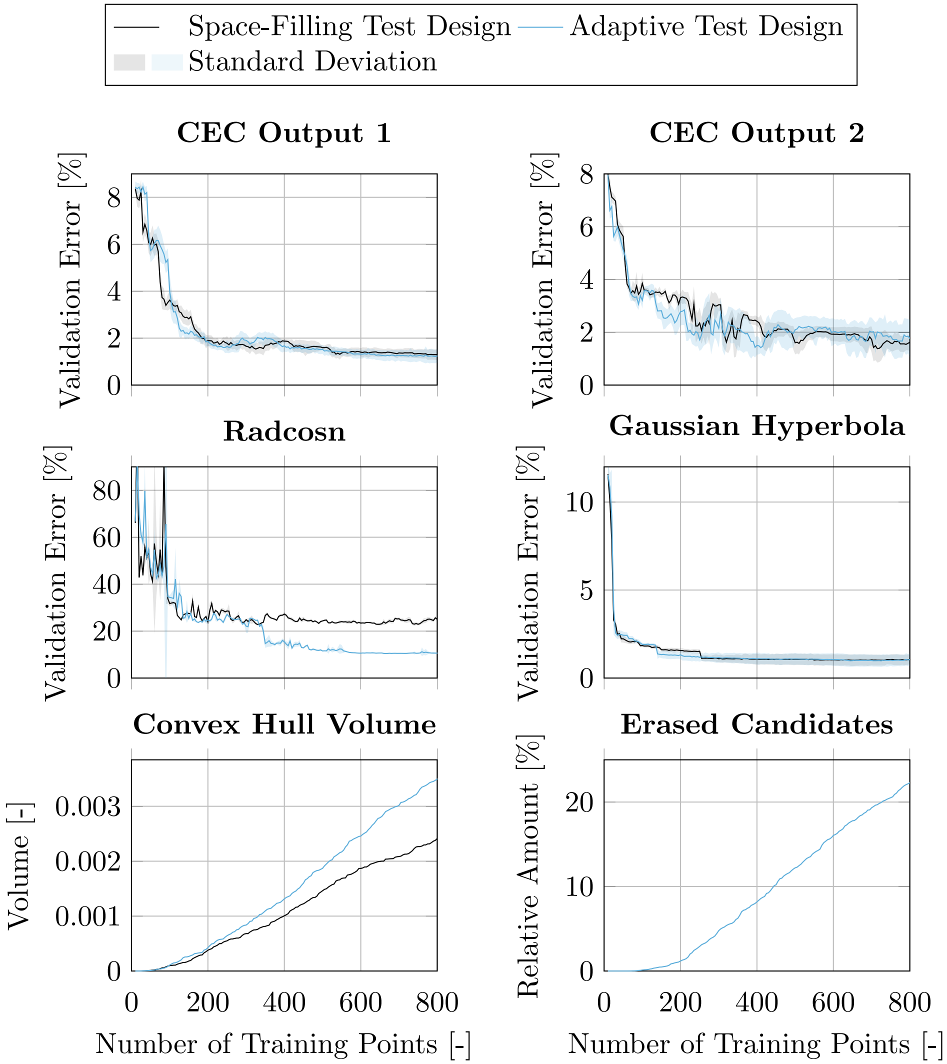


Figure 3.14: Top four plots: Relative validation error trend for the 4 artificial models CEC1, CEC2, radcosn and Gaussian hyperbolic function as shown in figure 3.9. 9 inputs are used for all models. The average validation RMSE is shown solid and the standard deviation between all 20 simulation runs is given as shadowed area for error stability estimation. Bottom left: Convex hull volume trend for the space-filling test design and the adaptive test design during the simulation of the artificial functions with 9 inputs. Bottom right: Relative amount of cumulated erased candidates for each iteration by using the adaptive test design method.

high dimension, a substantial improvement is achieved by the exclusion area strategy.

3.5 Chapter Summary

The main target of this chapter was to investigate a strategy that is able to identify, predict, and deal with engine boundaries during a test bench measurement with an adaptive test design method, where an amount of inputs more than 7 has to be considered. A boundary model and a boundary prediction for the non-known area was introduced to serve as the base for the adaptive test planning. The convex hull-based model and extension strategy answered the open questions how to deal with and achieve

- piecewise-defined limit behavior and
- a continuous measured input domain volume increase

and also involved the criterion to pay attention to more than one output, which is, however, not answered. Since the hull model only incorporates taken measurements in the input domain and their information about the boundary status, piecewise-defined limit behavior is representable. The output behavior is not incorporated, which is why output limits like engine misfire, knocking or other outputs with strong gradients could be included. This advantage is in contrast to output-driven boundary modeling strategies, which are hardly able to represent piecewise-defined limits.

The continuous measured input domain volume increase is achieved by a space-filling test design, where only test points within the predicted non-boundary area are taken into account. These areas are identified by the convex boundary criterion. Only parts outside the measured area that could possibly fulfill the convex hull criterion are taken into account. An underlying Sobol-based candidate set enables the maximin algorithm to find a next space-filling test point, where user defined convex and non-convex input domain restrictions are involved. Since the developed methods are geometrically based and rely on the convex hull criterion, no user configuration is mandatory for parameterization.

The developed test design strategy is validated by two different simulation models. A diesel engine simulation model with 7 inputs, based on a Gaussian process model, and a combination of four different artificial models with 9 inputs dealt as process models. Gaussian noise was added on the input and output and a Wiener process-based plant noise provided the necessary non-continuous system behavior during a measurement campaign. The simulation results were compared to a space-filling test design that was planned in advance by means of the same candidate set as applied to the adaptive approach. The criteria to evaluate were the resulting convex hull volume, which corresponds to the measured area, and the model validation error. The error was calculated as RMSE by a validation point set with 10000 measurements without noise within the true boundary area.

A significant increase of the convex hull volume of 45 % within the 9-dimensional test case with 800 measurements and nearly 100 % with 500 measurements within the 7-dimensional test case could be achieved. The validation error also exhibits a moderate to significant reduction, both during each measurement campaign and also with the maximum amount of measurements tested. Compared to the space-filling test design planned in advance, the validation error is not reduced for all outputs but does not increase for any output with using the adaptive approach.

The developed method contributes to a considerable reduction of measurements, with up to 42 % in the 7-dimensional simulation and up to 27 % in the 9-dimensional simulation, or an increase of model evaluation area and at the same time reduces the overall validation error. Compared to state of the art methods, the algorithms are able to incorporate piecewise-defined limit behavior and at the same time increase the measured input domain volume significantly.

4 Model-Based Test Design

In the preceding chapter, it was discussed how engine operating boundaries can be involved into the test design during an adaptive test design strategy. The resulting model quality could be optimized by a boundary estimation due to a more reliable measurement in terms of drivability of planned test points. In case no process information is present, a low-discrepancy test design is the most appropriate way to calculate a test plan for collecting information to train a GP model. Knowledge about the process behavior is generated during a measurement campaign that additionally can be included into the test point search during an adaptive test design. The quality of a low-discrepancy test design, calculated by the maximin definition (2.55) for example, is raised during the adaptive test design in case the test points cannot be applied as designed due to the occurrence of engine boundaries. However, this procedure does not incorporate process knowledge about the input-output mapping. Especially for strong nonlinear functions or a different influence of the input parameters on the output, a low-discrepancy test design is not the first choice. Some sequential test design strategies are already discussed in chapter 2. For instance, a sequential maximum entropy design can be applied within a GPM application. The process knowledge is incorporated by the entropy through the covariance function of the GPM. Since the hyperparameters of the squared exponential covariance function are updated, the covariance matrix changes in each iteration and the entropy change for a given test point location is different compared to a fixed hyperparameter assumption. This calculation is an easy method but does not deliver satisfying results for a general application (refer to section 2.4.4 and [Rai15]). Further strategies exist, see e.g. [Rai15; Xie16], but are not compared or even applicable in engine calibration. Additionally, no clear solution for the optimization of several outputs is given in literature.

The aim of this chapter is to identify a best-fitting solution for the adaptive test design, which improves model error reduction for one or several outputs compared to a low-discrepancy test design given the specified conditions. Three different new or modified implementations of test design methods are given and compared to state-of-the-art methods by a simulative investigation. The uncertainty-based test design (section 4.1), a relevance-based test design (section 4.2), and an adjusted MI test design (section 4.3) are explained. A simulation is set up and the results are discussed in section 4.4. The last section 4.5 summarizes the model-based test design chapter.

4.1 Uncertainty-Based Test Design

The GP model is able to predict the uncertainty at any input domain location by means of the prediction variance (2.38). Since this information is present, the test design is able to utilize it and place new test points where a high uncertainty is assumed. This method is very easy to apply and fast to calculate. However, it also has some disadvantages. The quality of this test design is strongly related to the model training quality. In case the hyperparameter estimation is of low quality the test point selection will result in a poor test design as well. In a worst case scenario the true process will never be identified correctly, because the test point selection prohibits an adequate hyperparameter training, which then results in a poor test point selection again. An example for this kind of misleading test design is shown in figure 4.1 in the upper plot. The aim is to identify the function $f(x, y) = x^4 + y^4$, where no noise is added to the function. A GPM is trained by means of cross-validation as described in section 2.1.4 after each iteration. The first 5 test points are given by the corners and the center location. A full factorial candidate set is used to select each next test point from. As can be seen, the GPM with maximum variance test design does not describe the influence of the input parameter y correctly (figure 4.1 bottom right). The influence of x is reproduced very well and a lot of measurements are planned at the border because the variance shows a high level at strong gradients.

Another problem with the variance-based test design is, similar to other output-driven test design methods, the dependency on only one process output. The variance of different outputs can be inverse proportional, leading to a high informative measurement for one output but a non-informative one for the other. For these two reasons a combined approach is being

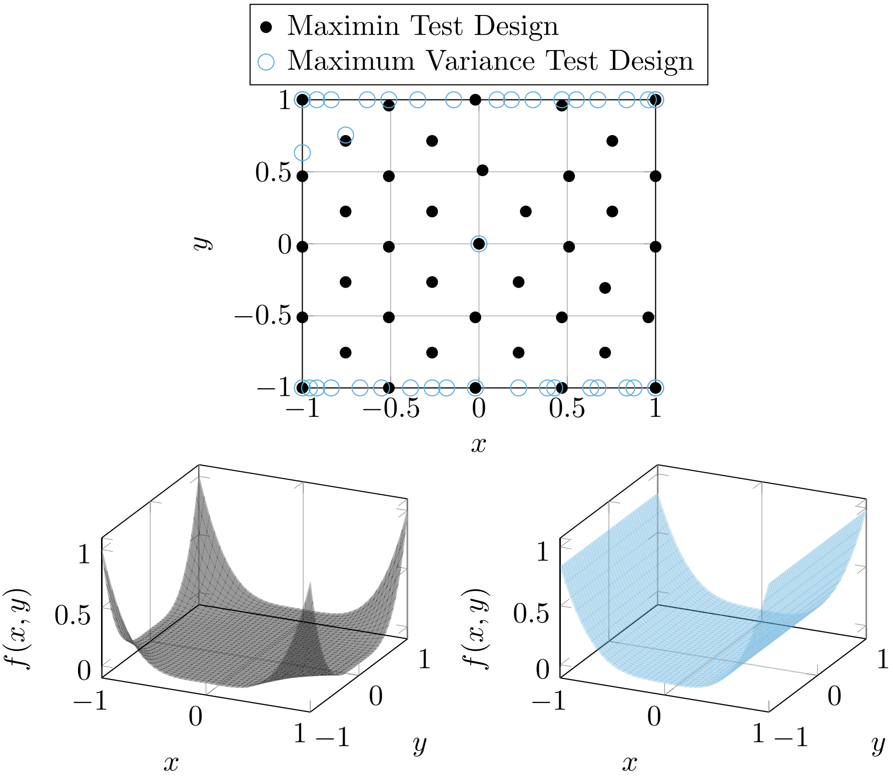


Figure 4.1: Top: Space-filling test design by maximin calculation and maximum variance test design for the function $f(x, y) = x^4 + y^4$ without noise. The GPM is trained after each new measurement and the latest model is used for the maximum variance identification. Bottom left: GPM regression with 40 space-filling training points. Bottom right: GPM regression with 40 maximum variance-based training points.

introduced that joins the benefits of both, variance-based and space-filling test designs.

The base of the combined approach is the maximin test design, applied to a given low-discrepancy candidate set $\boldsymbol{S}$ like a Sobol sequence. The uncertainty for each candidate is added to the distance criterion of maximin to achieve a combined rating for each candidate. A variance value is determined for all candidates by the current model of a given output by

$$\boldsymbol{v} = \mathrm{Var}_{\mathrm{GP}}(\boldsymbol{S}). \tag{4.1}$$

The distance and variance have to have the same domain for an equivalent rating. All uncertainty values are therefore normalized to [0,1] given the minimum and maximum variance, resulting in $\boldsymbol{v}'$. The same counts for the minimum distance of each candidate, given by the inner part of (2.55) to retrieve

$$\boldsymbol{d} = \min_{\boldsymbol{x} \in \boldsymbol{X}} d(\boldsymbol{x}, \boldsymbol{S}) \tag{4.2}$$

with already measured test points $\boldsymbol{X}$. The minimum distance vector $\boldsymbol{d}$ is min-max normalized to $\boldsymbol{d}'$ and the normalized variance values are added for the rating. The next test point then is identified by

$$\boldsymbol{x}' = \underset{\boldsymbol{S}}{\mathrm{argmax}}(\boldsymbol{d}' + \boldsymbol{v}'). \tag{4.3}$$

This procedure takes the variance of only one output into account. However, due to the incorporation of the discrepancy, this kind of test point generates information also for all other outputs. On the one hand, this test design is a compromise of a model-based and a uniform test design and will not generate as much information for one output as possible. On the other hand, it is a robust design, which will not result in a misleading measurement as given in figure 4.1. In the worst case, the incorporation of the variance will lead to the same result as if only the distance criterion is consulted. In best case, more information for the considered model is generated, where at the same time at least the same information is produced for all other models as with a strict space-filling design.

The combined test design aims at a process behavior that offers different gradients dependent on the input domain location. A high benefit is present if the output does change only slightly in large areas but changes rapidly in small areas. Such an extreme test case is defined by the function

$f(\boldsymbol{x}) = x_1^3 + x_2^3$ for $\{\boldsymbol{x} \in \mathbb{R}^{\mathrm{dim}>2} | -1 \leq \boldsymbol{x} \leq 1\}$. With dim $= 4$ a simulation is executed, where a relative Gaussian input noise with $\sigma_i = 0.02$ and a relative Gaussian output noise with $\sigma_o = 0.05$ are added. The simulation run is repeated 30 times to have a statistical valid result. The first 8 test points are given by the input domain center and 7 corner points. A full factorial grid with 18 grid points per input is used as candidate set and a grid with 32 grid points per input as validation set, which is not used for model training. The mean validation RMSE for 72 iterations is shown in figure 4.2. Whereas the space-filling distribution does not reach the low validation error level, the combined approach favors test points within the strong-gradient areas. This behavior results in a lower overall validation error, because the model is able to predict the process outcome within the high-gradient area more accurate. As the space-filling criterion is incorporated, the low-gradient area still is predicted at high precision. Additionally the combined approach reaches the low error level with less test points than the space-filling distribution. The variance criterion detects the high uncertainty in critical areas and favors a test point selection within these, leading to a faster model quality gain. Compared to the space-filling test design, the standard deviation of the model error is higher using the combined approach. This behavior reflects the variation of hyperparameter training, which results in a different test point selection. At some iterations, a slightly worse validation error can be reached, but in most cases a significantly reduced validation error is achieved. At 58 test points the combined approach results in a stable, high-quality model with nearly no difference between each simulation run.

In case several process outputs have to be modeled, a decision has to be made which model is consulted for uncertainty prediction. An adjusted round-robin procedure is suggested, where the worst models are favored. The model quality is not determined by the validation error only but also by means of a comparison of the worst RMSE to the measurement repetition error [IAV20]. The worst RMSE is the highest value of the training e_t, validation e_v and LOOCV RMSE e_l. This value is normalized to the standard deviation of all measurements $\boldsymbol{y}$ to

$$\nu_w = \frac{\max(e_t, e_v, e_l)}{\sigma(\boldsymbol{y})} \tag{4.4}$$

and represents the worst normalized error for the considered output. A

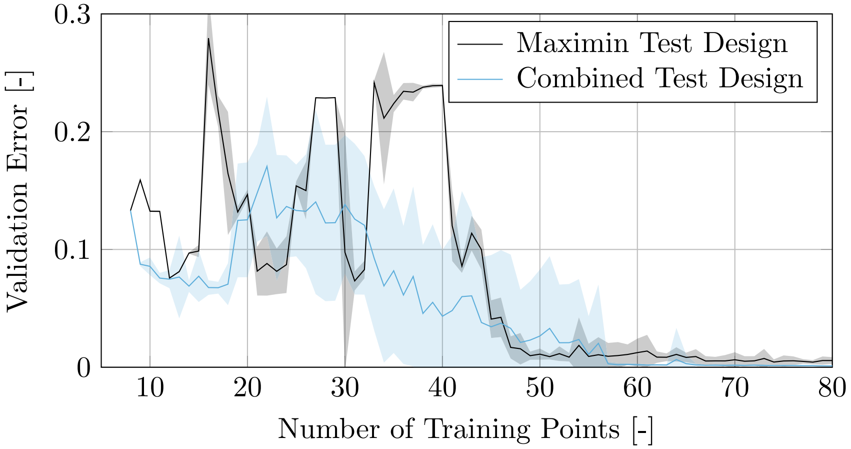


Figure 4.2: Validation error trend for a simulation with a space-filling test design by the maximin algorithm and the variance-based combined approach. The simulation is executed 30 times. For both methods, the mean (solid line) and the $\pm\sigma$ interval (shadowed area) is shown.

threshold is defined by all repetition measurements

$$\delta = \text{mean}(0.18k, \frac{\sigma(\boldsymbol{x}_{r,1})}{\sigma(\boldsymbol{y})}, \frac{\sigma(\boldsymbol{x}_{r,2})}{\sigma(\boldsymbol{y})}, ..., \frac{\sigma(\boldsymbol{x}_{r,k})}{\sigma(\boldsymbol{y})}) \tag{4.5}$$

for k different repetition settings $\boldsymbol{x}_{r,1}, ..., \boldsymbol{x}_{r,k}$. A status value is derived by classification for each model. The worst normalized model error ν_w is compared to the threshold δ in the interval

$$\text{status} = \begin{cases} 1 & \text{for } \nu_w \leq \delta \\ 2 & \text{for } \delta < \nu_w \leq 2\delta \\ 3 & \text{for } \nu_w > 2\delta. \end{cases} \tag{4.6}$$

Models with higher status values are preferred for variance evaluation to identify the next test point by the combined method. If only one test point is requested the model with highest status number is used. In case several models have the same highest status number, a model is selected randomly.

4.2 Relevance-Based Test Design

An adaptive test design algorithm has to find a next test point to gain as much information as possible for the training of a given model type. The GP model as discussed in section 2.1.4 is trained by hyperparameter selection given measurement data. Once selected, the hyperparameters, and especially the length scale values $\boldsymbol{l}^2$, present an assumption about the impact of each single input on the output. This influence can be utilized for a test point selection. Inputs with a low impact on the output do not necessarily have to be observed as detailed as inputs with a high impact.

The length scale $\boldsymbol{l}$ gives rise about how far one has to move in the input domain to significantly change the value of the function. Hence, the inverse of the length scale is proportional to the input impact. The incorporation of the length scale values is done by a space-filling test point selection in a distorted input domain. A Sobol-based candidate set $\boldsymbol{S}$ is distorted by the inverse of the length scale values

$$\boldsymbol{s}_d = \boldsymbol{l}^{-1} \circ \boldsymbol{s} \quad \forall \boldsymbol{s} \in \boldsymbol{S} \tag{4.7}$$

with the Hadamard or element-wise product $\circ$. Applying the maximin criterion on the distorted candidate set $\boldsymbol{S}_d$

$$\boldsymbol{x}'_d = \underset{\boldsymbol{S}_d}{\operatorname{argmax}} \min_{\boldsymbol{x}_d \in \boldsymbol{X}_d} d(\boldsymbol{x}_d, \boldsymbol{S}_d) \tag{4.8}$$

with the distorted, already measured test points $\boldsymbol{X}_d$ results in a new test point in the distorted domain. The back transformation

$$\boldsymbol{x}' = \boldsymbol{l} \circ \boldsymbol{x}'_d \tag{4.9}$$

gives the new test point $\boldsymbol{x}'$ that fulfills the relevance-based test design criterion.

As an example the function $f(\boldsymbol{x}) = 2x_1 + 3\sin(4x_2)$ with $\{\boldsymbol{x} \in \mathbb{R}^2 | -1 \leq \boldsymbol{x} \leq 1\}$ is considered, which is represented in figure 4.3. The input x_1 has a simple linear influence, while the input x_2 offers a sinusoidal correlation with short wavelength. It is easy for a model and especially for a GP model to identify the linear influence of input x_1, as long as two test points show a significant distance from each other in the x_1 domain. The input x_2 needs a well distributed test point design with a high resolution of that

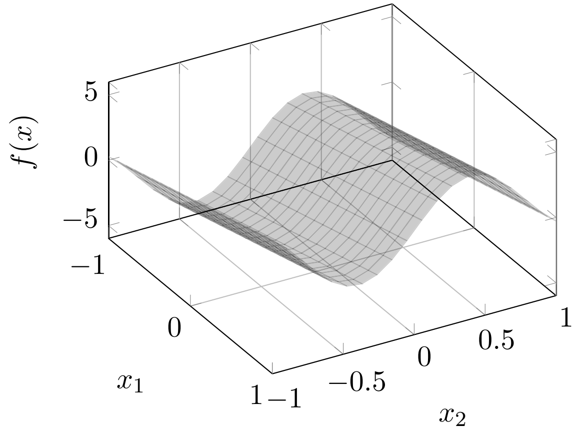


Figure 4.3: Artificial test function $f(\boldsymbol{x}) = 2x_1 + 3\sin(4x_2)$ in the input domain [-1,1]^2

input. Since the test function offers no interaction between both inputs, it is comparatively easy to identify and is only used to demonstrate the influence of a relevance-based test design. A simulation shows the benefit of the relevance-based test design. A space-filling distribution, calculated by the maximin algorithm, is compared to the relevance-based test design. The adaptive, relevance-based test design is generated by fixed relative length scales of $l_{rel,1} = 15$ and $l_{rel,2} = 1$, which are not changed during the simulation. Provided no prior knowledge about the process is given, this behavior is not applicable in a real test. The estimated length scales after each measurement have to be used instead of fixed relative length scales. However, it is chosen to demonstrate the relevance-based test design potential. A relative Gaussian input noise with $\sigma_i = 0.02$ and a relative Gaussian output noise with $\sigma_o = 0.05$ are added and the simulation is run 30 times for statistical validity. Figure 4.4 shows the validation error for both the space-filling and the relevance-based test design, calculated by 10000 grid-based validation points.

An obviously relevant gain in information about the process is generated by the relevance-based test design compared to the space-filling design. The test function is roughly identified 3 test points earlier than by the reference test design, which only is able to identify the process given 12 training points. A detailed identification is firstly present at 22 test points by the space-filling distribution, while the relevance-based design has a very low

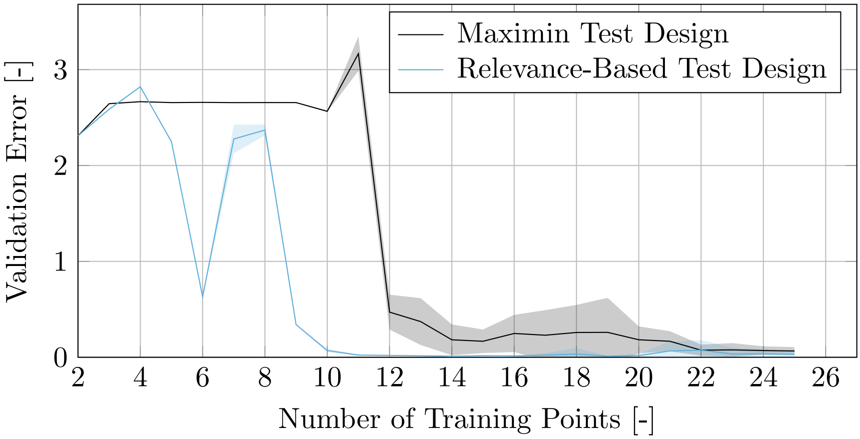


Figure 4.4: Validation error trend for a simulation with a space-filling test design by the maximin algorithm and the relevance-based approach. The simulation is executed 30 times. For both methods the mean (solid line) and the $\pm\sigma$ interval (shadowed area) is shown.

validation error already at 11 training points. Due to the high informative test design to identify the sinusoidal influence, nearly no deviation between the test runs is present. A contrary property is given with the space-filling distribution, which causes difficulties until 22 training points, noticeable by the higher standard deviation.

During an adaptive test design campaign the relevance-based algorithm is strongly dependent on the quality of length scale estimation. In case a high mismatch between the optimal and the estimated length scale is present, the test point distribution results in a very low quality design. In a worst case scenario, the true process is never identified, because the length scale estimation forces the relevance-based test design to find test points in non-informative areas again. To prohibit such a behavior, the maximum relative length scale is restricted by an upper bound. A relative length scale is calculated by

$$\boldsymbol{l}_{rel} = \frac{l_i}{\min(\boldsymbol{l})} \text{ for } i = 1, ..., \dim \qquad (4.10)$$

with the restriction of $\boldsymbol{l}_{rel} \leq 30$, resulting in a relative length scale domain of $[1, 30]$. The upper bound of 30 is chosen as a compromise between information gain and robustness.

In case several outputs need to be optimized a similar procedure as introduced in section 4.1 is applicable, considering only the model with lowest quality. However, a more integrated procedure shall be introduced to achieve a continuous optimization of all outputs other than with a round-robin or a batch optimization. The length scales of all models are combined to a mean length scale that is used for the relevance-based test design. The arithmetic mean of all relative length scales for input i is calculated

$$l_{rel,i} = \overline{l_{rel,i,1}, l_{rel,i,2}, ...l_{rel,i,k}} \tag{4.11}$$

for k outputs. The relative length scales again need to be normed and restricted to $[1, 30]$. A weighting to favor a test design for outputs with low quality is applicable by a weighted arithmetic mean function. For instance, the inverse quality status values (4.6) or a calibration engineer's rating could be used.

The space-filling test design in a distorted input domain allows a test design that contradicts the Latin hypercube design criteria. On the one hand, this is no criterion for the relevance-based test design, on the other hand, it provides a further improvement possibility if both criteria are combined. A test point calculated by the maximin definition in the distorted input domain is slightly moved by a local optimization in the following way. A loss function analyzes the dimension individual minimum distances to all given training points in the non-distorted domain by

$$\mathcal{L}(\boldsymbol{x}') = -\sum_{i=1}^{\mathrm{dim}} \min_{x_i \in \boldsymbol{X}} (|x'_i - x_i|) \tag{4.12}$$

with the measured test points $\boldsymbol{X}$. The minimization of the negative dimension specific distances aims at slightly shifting the given test point away from the measured ones in a 1-dimensional projection. Constraints are introduced to not deteriorate the relevance-based test design criterion and keep the test point within the input domain bounds. A nonlinear optimization constraint limits the loss of the space-filling criterion. Since the relevance-based test design rating is calculated in the distorted domain, the constraint is necessarily applied herein instead of the original domain. A loss of 5 % is permitted for the search. The aim of the optimization is not to find a globally different test point but to shift the given one slightly. Therefore, hard bounds keep the search area small. Half of the Euclidean distance to the closest training point

$$d_{max} = \max_{\boldsymbol{S}_d} \min_{\boldsymbol{x}_d \in \boldsymbol{X}_d} d(\boldsymbol{x}_d, \boldsymbol{S}_d) \tag{4.13}$$

is added or subtracted from the optimization start point $\boldsymbol{x}_0$, respectively, and used as bounds by

$$\begin{aligned} x'_{i,d} &\geq x_{i,0} - \tfrac{d_{max}}{2} \\ x'_{i,d} &\leq x_{i,0} + \tfrac{d_{max}}{2} \end{aligned} \quad \text{for } i = 1, ..., \text{dim}. \tag{4.14}$$

Additionally, the distorted input domain bounds

$$\begin{aligned} x'_{i,d} &\geq 0 \\ x'_{i,d} &\leq \tfrac{1}{l_{i,rel}} \end{aligned} \quad \text{for } i = 1, ..., \text{dim} \tag{4.15}$$

define an upper and lower constraint.

As an example, figure 4.5 is considered. A measurement at 5 different locations is given and a next test point has to be found for a next measurement. A simple space-filling test design computes a test point that neither respects an estimated influence of the input to the output nor considers the LH criterion. Assuming relative length scales of $l_{1,rel} = 3$ and $l_{2,rel} = 1$, the relevance-based test design applies the space-filling criterion in the distorted input domain, which is shown in the right plot. The horizontal axis shrinks due to the low relevance, while the vertical axis stays the same and is weighted significantly higher regarding the distance. Therefore, the space-filling test point calculated in the original domain does not fulfill the criterion anymore. A solution to fulfill the relevance-based criterion is a test point that mainly increases the density in the vertical domain. However, the relevance-based test point's horizontal location is exactly the same as two already taken measurements and gives a linear dependency in the vertical domain. To overcome this situation, the relevance-based point is slightly shifted, where the maximin criterion in the distorted domain is worsened marginally, while the dimension dependent distance to the measurements is significantly increased in the original domain.

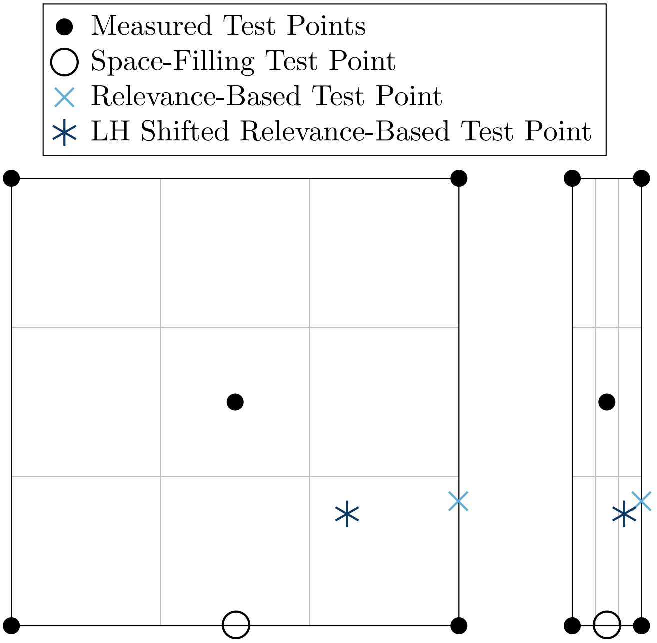

● Measured Test Points
○ Space-Filling Test Point
× Relevance-Based Test Point
∗ LH Shifted Relevance-Based Test Point

Figure 4.5: Demonstration of different test design strategies. Based on taken measurements, a next measurement is planned space-filling, relevance-based or relevance-based and shifted by LH criterion. The taken measurements serve as inclusions for each test design. Left: Test design in the original domain. Right: Test design in the length scale-based distorted domain with length scales $l_{1,rel} = 3$ and $l_{2,rel} = 1$.

4.3 Mutual Information

The mutual information criterion as described in section 2.4.4 is not used commonly in terms of adaptive test design within model-based calibration. Even though it reduces the disadvantages of the entropy test design, there is one main reason why it is used seldom and is difficult to apply. The computational complexity to find a next test point by the calculation of

$$\mathcal{I}(\boldsymbol{x}^*) = \frac{\sigma^2_{\boldsymbol{x}^*} - \boldsymbol{K}_{\boldsymbol{x}^*\boldsymbol{X}} \boldsymbol{K}^{-1}_{\boldsymbol{X}\boldsymbol{X}} \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{x}^*}}{\sigma^2_{\boldsymbol{x}^*} - \boldsymbol{K}_{\boldsymbol{x}^*\bar{\boldsymbol{X}}} \boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} \boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{x}^*}} \qquad \text{(2.65 revisited)}$$

for each potential test point $\boldsymbol{x}^*$ is very high and strongly depends on the candidate set size if an input domain discretization is the base. If used in an adaptive test design, the computational time to find a next test point is crucial and limitations have to be met. That is, a solution has to be investigated to enhance the test point search speed by the MI criterion. Since the MI cost function always has several local maxima, a global search is indispensable. A low-discrepancy candidate set seems to be favored over an exhaustive global optimization. The amount of candidates can easily be regulated, which is directly proportional to the calculation time. Additionally, a candidate set is strictly necessary to calculate the denominator in (2.65).

One possibility to enhance the calculation duration without the reduction of potential next candidates is to lower the computational effort for the entropy estimation regarding the unknown region in the denominator. As discussed, the most calculation expense is generated by the inverted covariance matrix of unmeasured candidates. If the size of the covariance matrix $\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}$ is reduced, the MI calculation duration for each candidate can be improved significantly. A Sobol sequence is used as candidate set, where a reduction of candidates does not necessarily lower the input domain coverage. A possible solution is to use the first n candidates without those already measured. The mutual information still is calculated for the whole given candidate set but with a reduced amount of candidates in $\bar{\boldsymbol{X}}$. The unknown region entropy estimation suffers from a candidate reduction, however, which has to be investigated.

A more coordinated method is the estimation of the unknown region entropy by a specifically designed candidate set. A space-filling test point distribution seems to be a reasonable solution, where already measured training points are used as inclusions during the calculation. This distribution could be more reliable for the estimation of entropy regarding unknown regions compared to a non-designed distribution. As an example consider figure 4.6. A measurement is performed at 10 locations in a 2-dimensional input domain. The base for the estimation of the unknown region entropy is given by 20 candidates, fulfilling the space-filling criterion for the given measurements. This procedure is very different to simply use the candidate set, because $\bar{\boldsymbol{X}}$ could completely change with each iteration. The performance has to be tested within a simulation environment and compared with the simple candidate-based solution.

Both process models as described in section 3.4.2 are tested for their sensitivity to a candidate reduction within the MI criterion. A variation of the

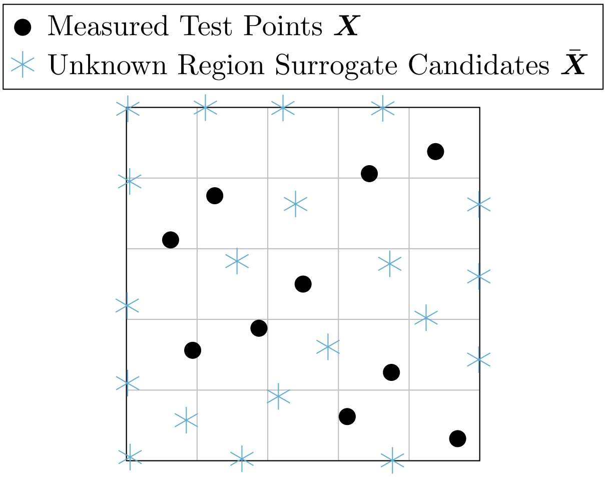

● Measured Test Points $\boldsymbol{X}$
✳ Unknown Region Surrogate Candidates $\bar{\boldsymbol{X}}$

Figure 4.6: Example situation for a mutual information calculation. 10 measurements are performed and a space-filling distribution is utilized for the entropy estimation regarding unknown regions in (2.65).

amount of candidates for both strategies is applied for the estimation of the unknown region entropy. The test cases offer 9 or 7 inputs, respectively, a relative Gaussian input noise with $\sigma_i = 0.02$, a Wiener process noise with $\sigma_w = 0.0005$ and a relative Gaussian output noise with $\sigma_o = 0.01$, where the noise is applied as defined in (3.13). Each simulation is run 5 times. A candidate set of 30000 Sobol distributed points in the unit input domain is the base for the iterative test point selection. The simple candidate reduction method selects the first n candidates to calculate the denominator covariance matrices. For calculation duration reasons a maximum of 2000 candidates is examined for the simple candidate reduction method. The space-filling unknown region entropy calculation utilizes the full candidate set to select candidates from, where a maximum amount of 2000 points is gathered in each iteration. The maximum amount of 2000 surrogate points is applied because it already is very expensive in terms of computing time. A higher amount of surrogate points would likely exceed the maximum duration for a test point search, dependent on the test case, as will be discussed. A validation error is calculated as the RMSE between model prediction and measurement given 10000 space-filling distributed validation points and normed by (3.14).

Figure 4.7 shows the validation error decrease result for the radcosn output during the artificial model simulation. Each line expresses the mean value over all 5 simulation runs. The adaptive test design only focuses the radcosn function output for all shown adaptive test design procedures. The space-filling distribution is calculated in advance in the normed, non-distorted domain. The first 10 test points are the same for all methods. The upper diagram shows the effect of the mutual information test design with the simple candidate-based procedure. As expected, the validation error highlights a strong relationship to the amount of training points. However, a distinct benefit in a candidate increase for the unknown region entropy estimation is not directly observable. This contradicts the assumption, that a simplification of the unknown region entropy calculation will lower the quality of the test design. Even the space-filling-based unknown region entropy method in the bottom diagram does not show an increase in model quality during the whole simulation compared to the simple candidate selection mutual information-based method. A high difference is apparent compared to the space-filling test design though. Independent of the unknown region entropy estimation, all methods highlight a significant increase in model quality during the whole simulation.

A different point of view is given by figure 4.8. Since there is no big difference between the simple method and the space-filling-based MI method, the space-filling method is shown only. The main statement of this graph is the possible reduction of the amount of necessary measurements by using the MI test design instead of a non-adaptive space-filling design. The horizontal distance in figure 4.7 between each MI simulation run and the space-filling test design simulation is plotted. The reference is the MI validation error, meaning the validation error at 150 training points with the MI test design with 1500 space-filling candidates is reached over 150 points later, at 300 test points, with the space-filling test design. No data is present for more than 394 training points, because the level of validation error is never reached by the space-filling test design within the simulation. It is important to consider the point deviation to judge the effective time reduction by using the adaptive test design. If the different amount of space-filling candidates are compared in this view, an explicit best solution could neither be given. An amount of 500 candidates seems to perform slightly worse between 300 and 400 training points, but until 300 measurements all point deviations are nearly the same. The same representation is given for the

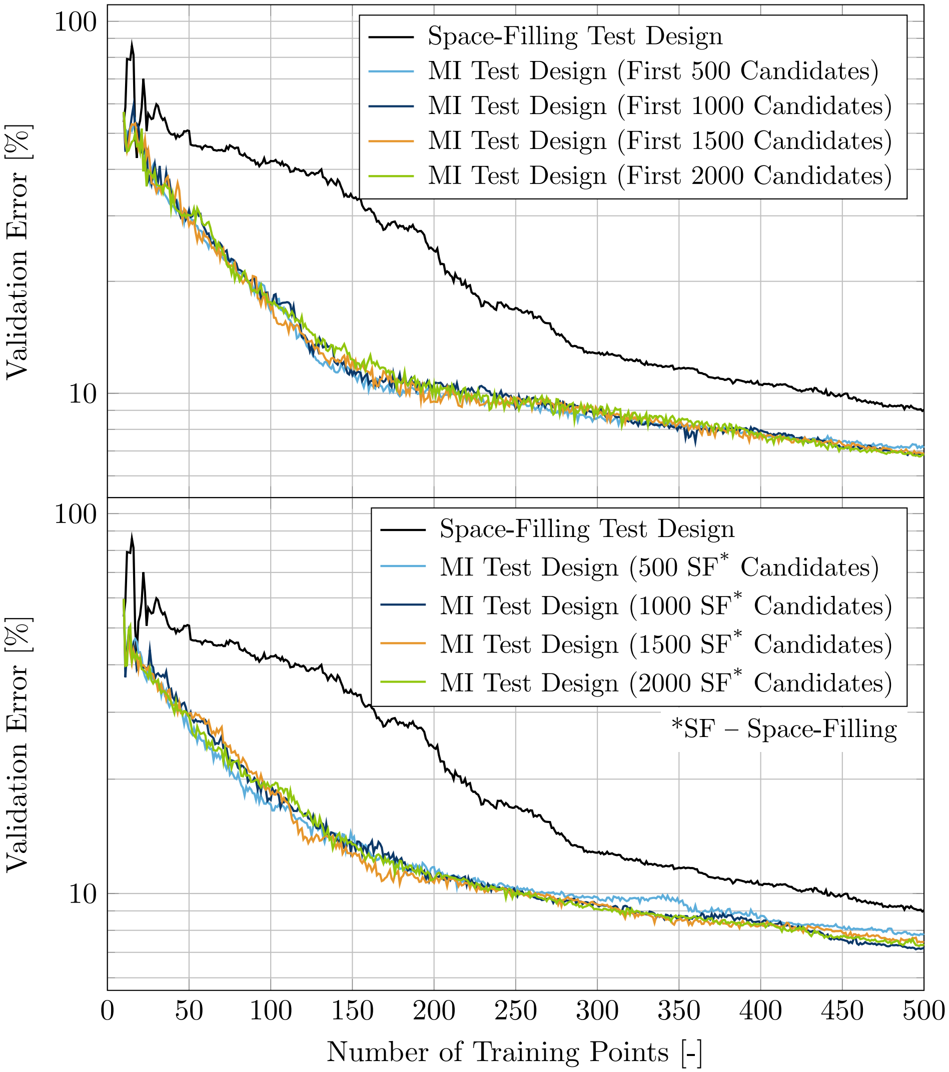


Figure 4.7: Logarithmic represented validation error trend for the mutual information test design with different unknown space entropy estimations. Each line describes the mean value from 5 simulation runs for the model output radcosn. The space-filling test design is shown as reference for a non-adaptive method. Top: MI test design with fix Sobol-based unknown region surrogate candidates. Bottom: MI test design with space-filling unknown region surrogate candidates calculated by maximin in each iteration.

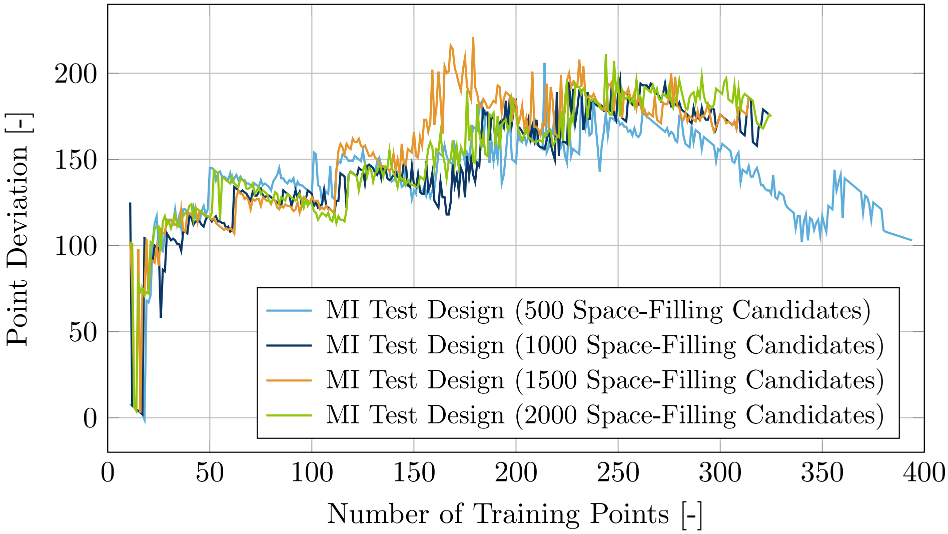


Figure 4.8: Point deviation for different mutual information test design validation errors for the radcosn output function within the 9-dimensional artificial model simulation. A positive value corresponds to the amount of test points that is reduced by MI compared to a non-adaptive space-filling test design to reach the same validation error. The distance can be thought of as the length of a horizontal line in figure 4.7 from each validation error value starting at the MI test design error to reach the space-filling test design error.

CO output of the 7-dimensional simulation in figure 4.9. Following a slight negative value in the beginning of the simulation, a significant reduction of measurements is achieved by each amount of candidates as well. Only an amount of 1500 candidates shows a worse performance than all other methods during the first 180 measurements. Contrary to the 9-dimensional test case, an amount of 500 space-filling candidates seems to be the best solution. However, the effect of a candidate reduction is different than assumed and a low amount of candidates to estimate the unknown region entropy seems to be reasonable as well.

To rate all different candidate reduction strategies within both simulations, two different metrics are introduced. As first metric, the area under the curve (AUC) criterion is applied to the validation error results. The lower the area under a validation error trend is, the better the model quality improvement capability is. The second criterion is defined by the regression coefficients of an exponential function. A least squares estimate of the

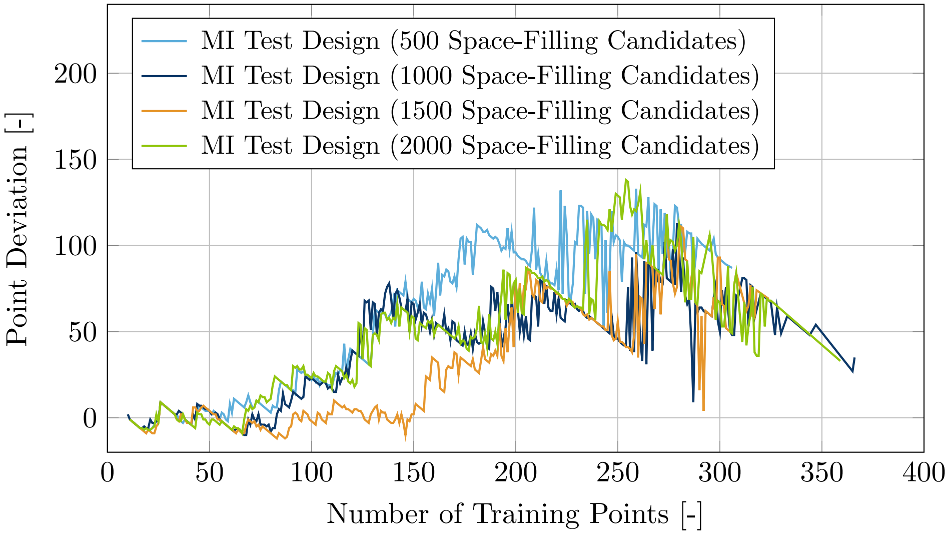


Figure 4.9: Point deviation for different mutual information test design validation errors for output CO within the 7-dimensional diesel engine GP model simulation. A positive value corresponds to the amount of test points that is reduced by MI compared to a non-adaptive space-filling test design to reach the same validation error.

function

$$f(x) = e^{a\log(x)+b} \tag{4.16}$$

trained to each trend gives the regression parameters a and b. The lower their value, the faster the validation error reduction is. Most important is the coefficient a, which explains the decrease rate with the amount of training points, where b is only an offset to the exponent. However, b is comparatively low in all test cases but has an impact on the performance. It can be seen as a multiplicative factor to the exponential decrease. Therefore, the sum of both parameters is taken to judge the quality. The result for both simulations is shown by a ranking in figure 4.10. The ranking is based on the AUC and $a + b$ data with a lower rank given for lower particular values, meaning the lowest rank equals the best performing method. The full data is given in table D.1 in appendix D. Both criteria show nearly the same rating within the 7-dimensional simulation and identify the space-filling-based MI design as the best method with still no clear solution to the amount of candidates but a very bad quality for 1500 space-filling can-

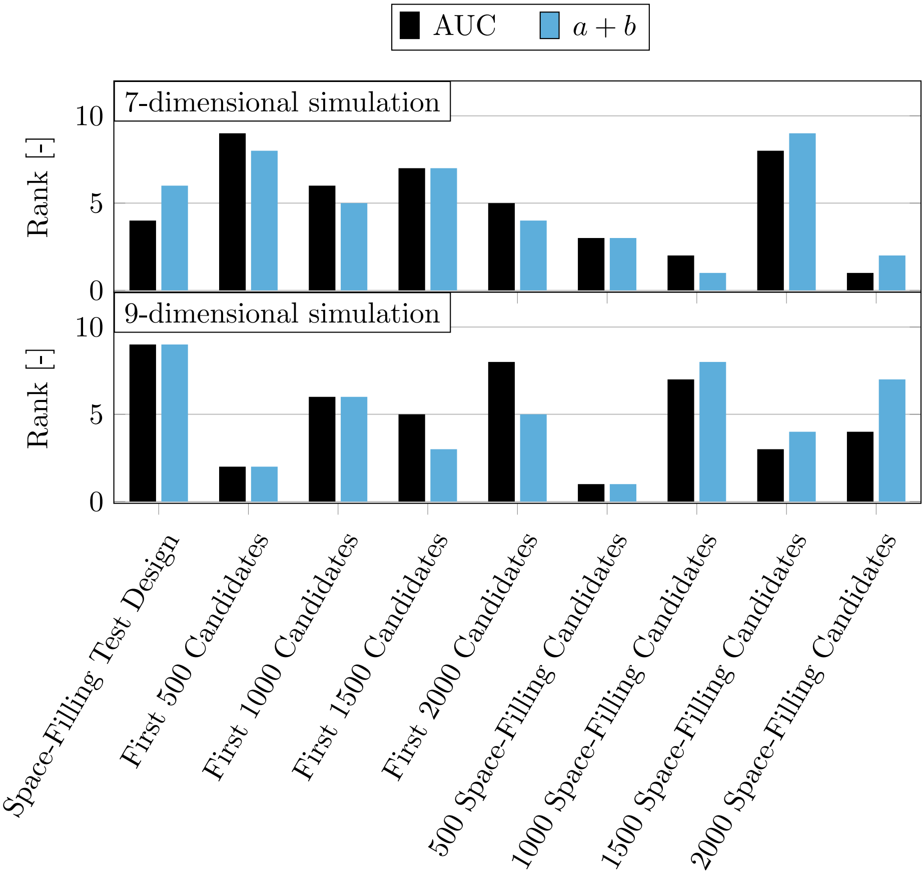


Figure 4.10: Rating for different MI-based test design strategies and a non-adaptive space-filling test design. The ranking for both rating methods, AUC and $a + b$, is based on the results shown in table D.1 in appendix D. A rank of 1 equals the best, rank 9 the worst performing method. Top: Results of the 7-dimensional diesel engine simulation for the output CO. Bottom: Results of the 9-dimensional artificial model simulation for the output radcosn.

didates. A similar situation is present for the 9-dimensional test case with a surprising performance of the simple method with 500 candidates and a poor performance of the space-filling-based MI with 1000 candidates. The amount of candidates regulates the tendency for a next test point to be planned near the boundary. As described, the maximum entropy design tends to plan test points mainly near the input domain boundaries. If the amount of candidates to calculate the unknown region entropy is reduced the design converges to a maximum entropy design. The variation of the unknown region candidate amount gives rise to a low amount of candidates being sufficient to reduce the tendency of a mostly near-boundary design. The best mean performance is therefore given by the space-filling planned MI design. A low amount of candidates with this test design seems to be more robust and provides good results in both simulations.

Another important criterion to judge the applicability of the mutual information algorithm is the duration to calculate a next test point. During the adaptive test design a fast calculation is necessary for a continuous measurement procedure. The present simulation is executed on a modern desktop computer with an Intel® Core™ i5-8500 3 GHz processor and 32 gigabyte RAM. The duration for the calculation of the last test point is given in figure 4.11 for the different amount of candidates in $\bar{\boldsymbol{X}}$ for both methods within the 9-dimensional radcosn simulation. As expected, the calculation duration increases exponentially due to the matrix product and matrix inversion in the denominator in (2.65). The space-filling-based procedure takes slightly longer, because the matrix $\bar{\boldsymbol{X}}$ has to be computed by maximin instead of a simple selection of the first n candidates. However, the difference is low compared to the absolute calculation duration. With each simulation run the calculation duration increases exponentially as well. The absolute cost is irrelevant in this example, because only the amount of already measured test points defines the matrix inversion cost of $\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}^{-1}$ in the nominator of (2.65). The maximum amount of test points is restricted to 500 in this example and contributes to the calculation duration, but is unnecessary for the relative comparison. However, if the method is applied to a test with a higher amount of necessary test points, the exponentially increasing calculation duration has to be considered. The application of 500 space-filling candidates is strongly favored compared to a higher amount. In case the MI has to be computed for several outputs, a calculation duration of over 40 seconds is not acceptable. The increase in calculation duration

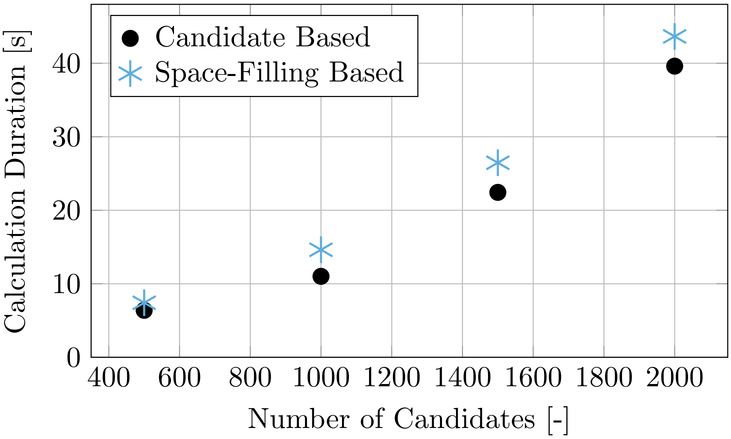


Figure 4.11: Average calculation duration for the 500th test point by mutual information within the 9-dimensional simulation in dependency of the amount of candidates in $\bar{\boldsymbol{X}}$ used for the unknown space entropy calculation

to find the space-filling candidates is very low and could be applied if only 500 candidates are used. Since the different amount of candidates does not offer a significant increase in model quality and a robust solution is given if 500 space-filling points are applied, this method is judged to be the best compromise for the following investigations.

To apply the mutual information criterion to a process with multiple outputs different methods are possible. A solution to gather as much information as possible for all outputs is to calculate a mean information content for each candidate. The mutual information for each candidate and each model is computed first. Then the candidate with highest averaged information content for k models

$$\boldsymbol{x}'_{MI} = \underset{\boldsymbol{s}\in\boldsymbol{S}}{\operatorname{argmax}} \sum_{i=1}^{k} \mathcal{I}_i(\boldsymbol{s}) \tag{4.17}$$

is selected as next test point. A weighting or a normed mean information content could also be applied, if a calibration engineer's preference is present.

4.4 Simulation-Based Investigation

The benefits of the algorithms developed in this chapter have to be tested and investigated regarding their effect within a measurement campaign. Similar to the simulation-based investigations in chapter 3, the model-based test design algorithms are tested in a simulation environment, which configuration is described and which results are discussed in this section.

4.4.1 Objectives

The effect of the introduced algorithms on a real engine identification process is unknown. The assumptions have to be proven, where the aim is to gain most possible knowledge about the process with the smallest amount of measurements. The influence of engine boundaries on the design is not considered here. An isolated test on model quality enhancement is executed in a unit hypercube input domain without restrictions. The procedure to judge the model quality is the application and consultation of a uniformly-distributed validation point set with 10000 test points. The validation RMSE calculated by model prediction and true process outcome rates the averaged overall model quality.

The algorithms are compared to a model-independent space-filling test point distribution calculated in advance of the measurements. Since the relevance-based test design with added LH criterion is a combination of two test design methods, an additional test design method is used for comparison. A model-independent space-filling test point distribution is the base for a LH optimized distribution. Each maximin calculated design point is shifted by the same optimization algorithm as discussed in section 4.2. A loss of 5 % maximum distance in the original domain is permitted. The constraints and optimization procedure are the same as defined for the LH optimized relevance-based design. The maximum entropy test design often utilized in adaptive test design strategies is simulated additionally for comparison.

The results have to be compared mainly regarding their ability to quickly reduce the validation error. This error type rates the generalization ability within the input domain, which is the main objective of the model-based test design introduced here. However, not only the resulting model quality with a defined number of iterations has to be conducted, but the error decrease trend over iterations is a necessary criterion to find a solution that reduces the error at each measurement stage. Commonly, the available measurement time at engine test benches is low, which is why the necessary amount of measurements to reach a model error saturation is hardly been given.

4.4.2 Simulation and Noise Model

The process models for the comparison of the introduced adaptive test design procedures are the same as introduced in section 3.4.2. Both the artificial models as well as the diesel engine simulation model serve as test functions. The focus of the simulation is to assess each method's influence on model quality. Therefore, the model boundaries are not considered except the minimum and maximum values for each input. The test design is planned within a unit hypercube and all measurements are taken as they are planned without respect to any boundary restrictions.

The application of a noise model is necessary for this investigation as well. Different test designs have very diverse impact if noise is applied or not. An i.i.d. Gaussian input noise, Wiener process-based plant noise and i.i.d. Gaussian output noise is applied as described in section 3.4.2. The input noise is considered with $\sigma_i = 0.02$, the Wiener process is generated with the length of the particular simulation measurement amount and $\sigma_w = 0.0005$ and an output noise with $\sigma_o = 0.01$ is applied.

Each simulation is repeated 10 times for both the artificial and the diesel engine simulation model. The adaptive test designs each utilize their introduced method to find a test point that optimizes all outputs simultaneously. Within the adaptive maximum entropy design, the test point with highest average entropy increase is used. A 7-dimensional test with 400 measurements is investigated with the diesel engine simulation and a 9-dimensional test with 500 measurements with the artificial test functions. Compared to the adaptive test design simulation (refer to chapter 3), the amount of tests in the 9-dimensional test case is reduced for simulation duration reason. Each test design strategy provokes a model training after each measurement, leading to very high computation times. The simulation was performed on a computer with two Intel® Xeon® E5-2680 v3 CPUs with 12 cores each. Nonetheless, a simulation run took around 36 hours for the 9-dimensional test, which is why the maximum amount of test points was reduced.

4.4.3 Discussion

The tests are judged by a normed validation error, calculated by (3.14). Figure 4.12 shows the error development for the diesel engine simulation. Regarding the general model error the soot emission output stands out with the highest and non-converged model error after 400 measurements. None of the test design methods is able to reduce this effect considerably. However, there are differences between all methods for each output. Especially within the first drop of validation error until 150-200 measurements the MI test design performs clearly best with the exception of the soot mass flow output. Overall the MI criterion and the relevance-based test design perform very well, while the performance of all other test design strategies differs from output to output. A better assessment is achieved, if the AUC criterion is considered to compare the strategies objectively. For each output and each averaged validation error trend, the area is calculated and the normed deviation to the space-filling test design is determined for easier comparison. A desired outcome should be as low as possible, which gives the lowest area and therefore lowest validation error during the whole measurement process. The AUC values for the 7-dimensional simulation are shown in figure 4.13. The diagram illustrates the relative deviation to the non-adaptive space-filling test design with a negative value showing an AUC reduction. The MI test design delivers the best overall performance with a reduction of 3–15 % compared to a non-adaptive space-filling test design. The most improvement for a single output is given by the relevance-based test design for the exhaust temperature output. However, this test design performs worse for the soot emission output, where the non-adaptive space-filling LH design performs best. The maximum entropy also shows good results for all outputs but is worse compared to the MI test design. A disappointing performance is given by the variance-based test design, which performs worse than the space-filling design for three outputs and worst within the adaptive approaches for nearly each output with only a single exception.

The analysis of the 9-dimensional artificial model allows a more detailed differentiation of the investigated test design methods, because their impact on the validation error trend is higher. The normalized validation error trend is given in figure 4.14 for all 4 model outputs. Especially for the radcosn function a very diverse error trend is visible during the measurement, while the error level after 500 measurements is nearly the same

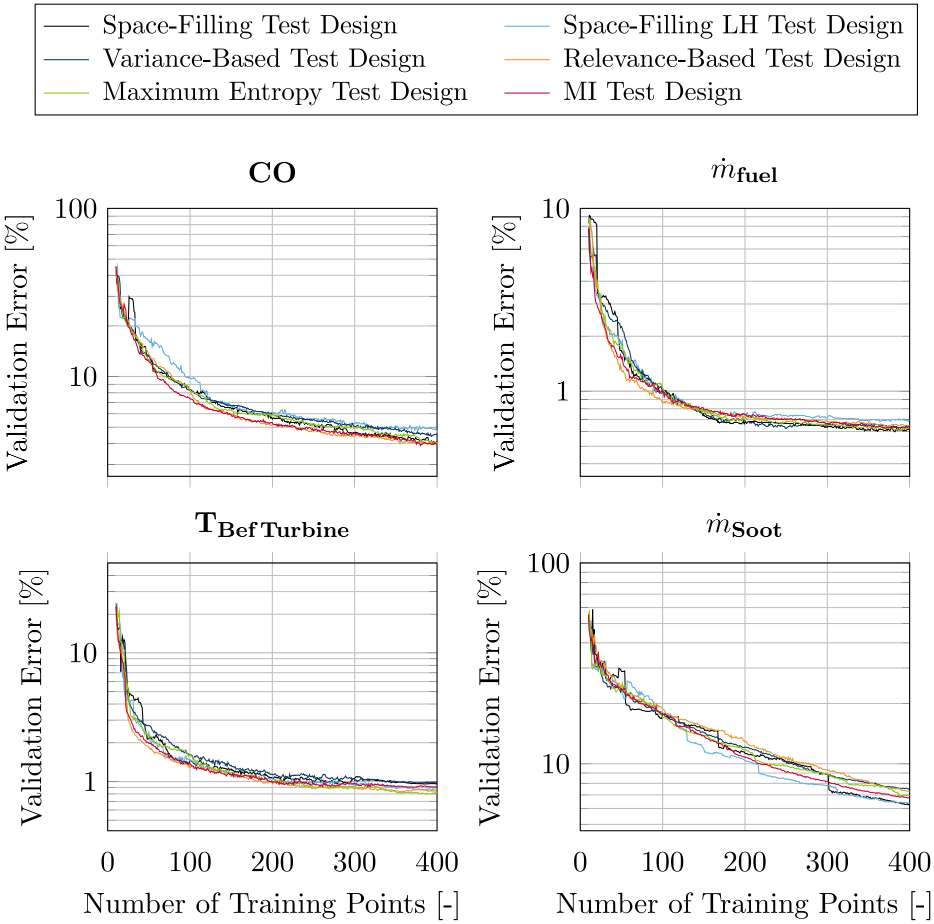


Figure 4.12: Comparison of different test design methods by the validation error decrease for the four outputs of the diesel engine simulation model with seven inputs. The simulation is run 10 times and the validation error is averaged and normed to the mean plus standard deviation of all validation simulations.

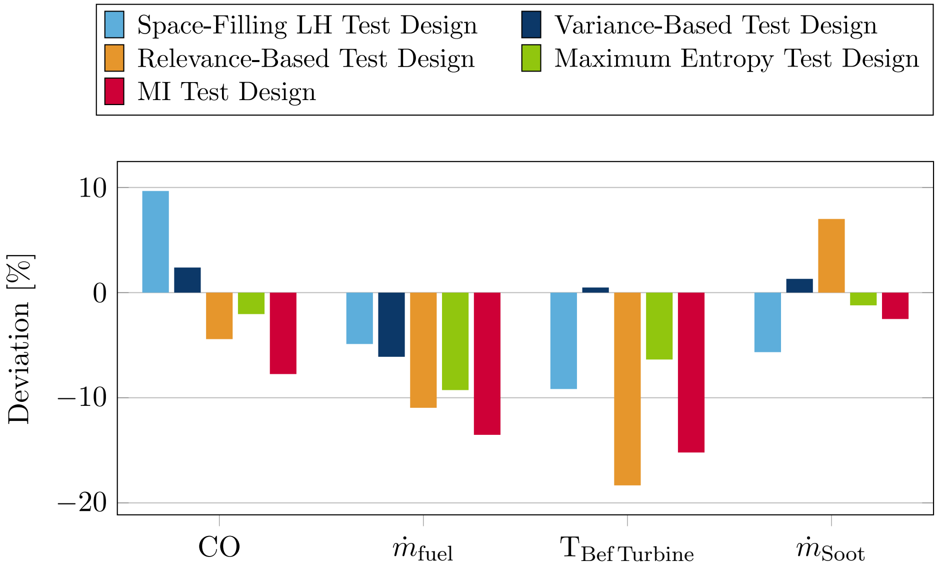


Figure 4.13: Relative AUC value for each test design and each output within the 7-dimensional diesel engine simulation. The relation is given as the relative deviation to the space-filling test design as reference for each test design method. A negative value defines a smaller area and therefore improves the validation error compared to the space-filling test design.

for all methods. The absolute error level is considerably higher than for all other outputs, which makes it most important to find a good test design that is able to reduce the error level as quickly as possible. The error trend is very different for the Gaussian hyperbola function, with which the error during the first 100 measurements is nearly the same for all methods but a clear differentiation between all methods is visible from that point on. Both CEC outputs show a different behavior for all methods during the whole simulation.

The MI test design clearly outperforms all other methods for all outputs. This behavior is underlined by figure 4.15, where the relative AUC deviation to the space-filling test design is plotted for all methods and each output. The relevance-based test design shows a slightly worse performance, which is surprisingly only a little better than the space-filling LH test design. However, the effect of the relevance-based part of this test design is visible by comparing it to the space-filling LH design. A very bad

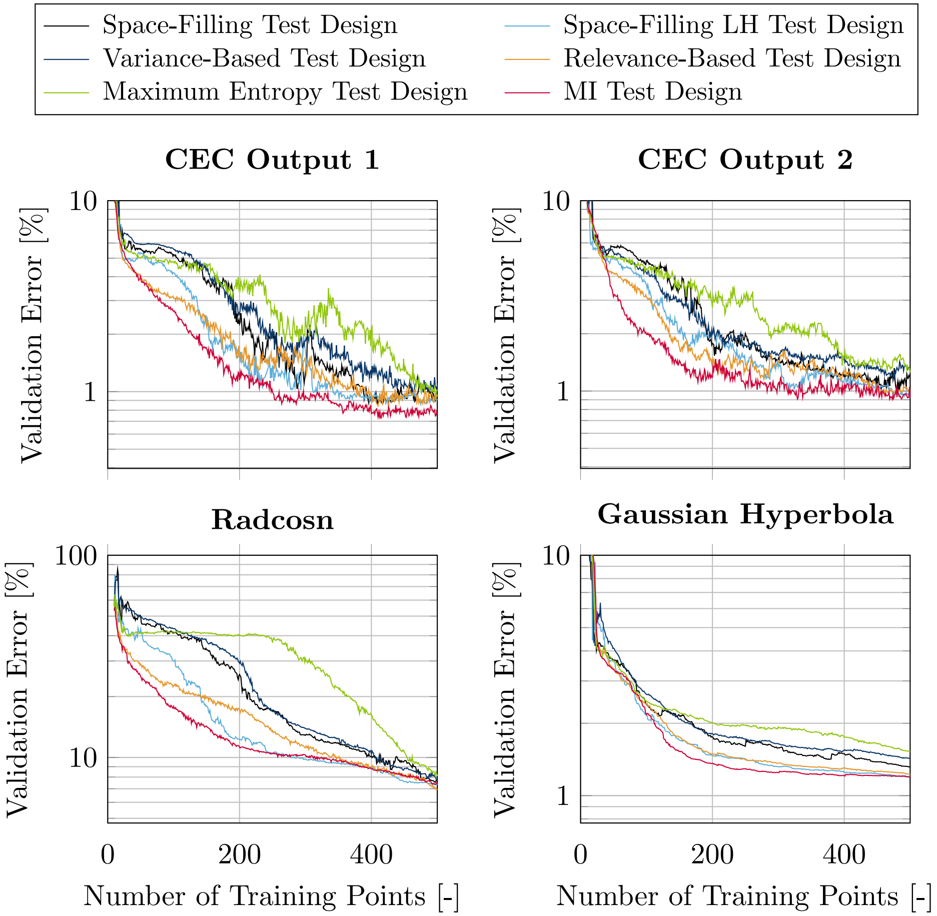


Figure 4.14: Comparison of different test design methods by the validation error decrease for the four outputs of the 9-dimensional artificial model. The simulation is run 10 times and the validation error is averaged and normed to the mean plus standard deviation of all validation simulations.

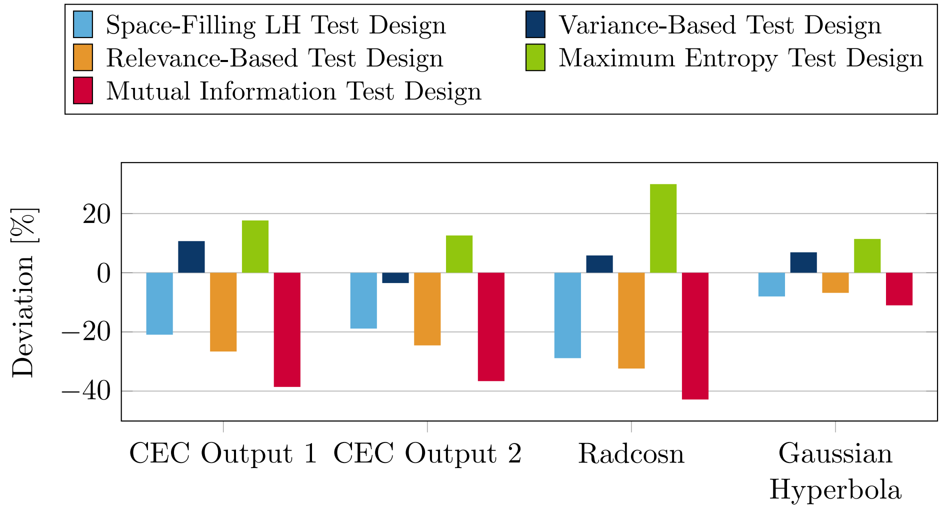


Figure 4.15: Relative AUC value for each test design and each output within the 9-dimensional artificial model simulation. The relation is given as the relative deviation to the space-filling test design as reference for each test design method. A negative value defines a smaller area and therefore improves the validation error compared to the space-filling test design.

performance is given by the maximum entropy design, which is significantly worse than the non-adaptive space-filling test design for all outputs. The variance-based test design shows a similar, non-satisfying result as with the 7-dimensional simulation. From a validation error performance point of view, the MI test design outperforms all other test designs, especially during the error decrease phase but also at error saturation for the artificial models. However, this test design is the most computationally intensive design. The relevance-based test design offers a much lower computational cost at a comparative but minor worse error decrease performance. The mean normed validation error deviation relative to the space-filling test design over both simulations results in a 21 % better performance for the MI test design, a 15 % better performance for the relevance-based test design, and an 11 % better performance for the LH space-filling design.

Another important criterion to judge the test design is the fluctuation between each measurement campaign. Since 10 test runs are performed, an analysis of the different error trends is possible. To simplify the comparison, box plots are derived from the simulation result. Figure 4.16 shows box plots for the AUC criterion for each output and each test design method within the 7-dimensional simulation. The relative deviation to the median of the space-filling test design result is given for each method. The expectation that the box size as well as the range of values is lowest for the non-changing space-filling and space-filling LH design is not fulfilled for each output. This means that an adaptive approach can also improve the model quality outcome robustness. The MI criterion offers a very small quantile for all outputs but with a high range for the output CO. However, even the worst run performs better than the median of a space-filling design. Except for the soot emission output, the relevance-based test design offers low box sizes and the lowest overall for the exhaust temperature. The variance-based and the maximum entropy design do not stand out, neither to be very fluctuating nor to be very stable. The MI design outstandingly performs in the 9-dimensional test case regarding the fluctuation between each simulation run, as can be seen in figure 4.17. For each output, it offers the smallest box, highlighting this method as a notably stable one. The bad performance of the maximum entropy design is even worsened by its high fluctuation. A similar result is present for the variance-based test design, showing the highest overall fluctuation. The relevance-based test design offers less variation than the non-adaptive space-filling design and a similar one to the space-filling LH design, which also performs well regarding the variation between the simulation runs. Concluding the fluctuation analysis, a similar result is present as the mean value analysis showed. The MI criterion is exceedingly robust beside its good overall performance. The relevance-based test design provides well and robust results, albeit worse than the MI criterion from both points of view. The adaptive maximum entropy and variance-based method perform worse than the non-adaptive space-filling LH design.

It strongly depends on the computer hardware and further necessary computations, which test design should be applied. The relevance-based test design is applicable to each standard hardware, because the computational cost is hardly higher than for a space-filling test design but it yields much better performance. The MI test design offers the best performance but for a very high computational cost. A decision has to be made in an overall examination of the requirements. In terms of this research, the consideration is given in the subsequent chapter 5, where a combined test design approach is developed to meet all objectives.

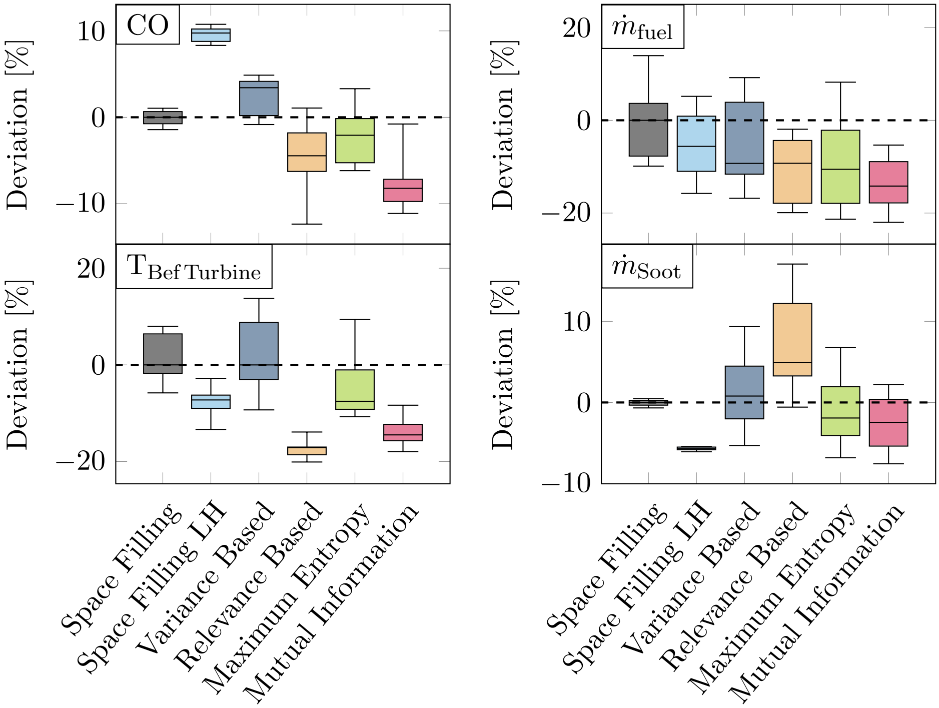


Figure 4.16: Box plots for the AUC value for each test design and each output within the 7-dimensional diesel engine simulation. The AUC values show the relative deviation to the median of the space-filling result. A negative value defines a smaller area and therefore improves the validation error compared to the space-filling test design. The whiskers reach from the minimum to the maximum value, which is why no outliers are present.

4.5 Chapter Summary

Within this chapter, different new and modified adaptive test design strategies were introduced to reduce the test error of a GPM as fast as possible and to identify a given process with highest possible accuracy. The three introduced methods were

- a variance-based design,
- a relevance-based design and
- an entropy-based modified mutual information design.

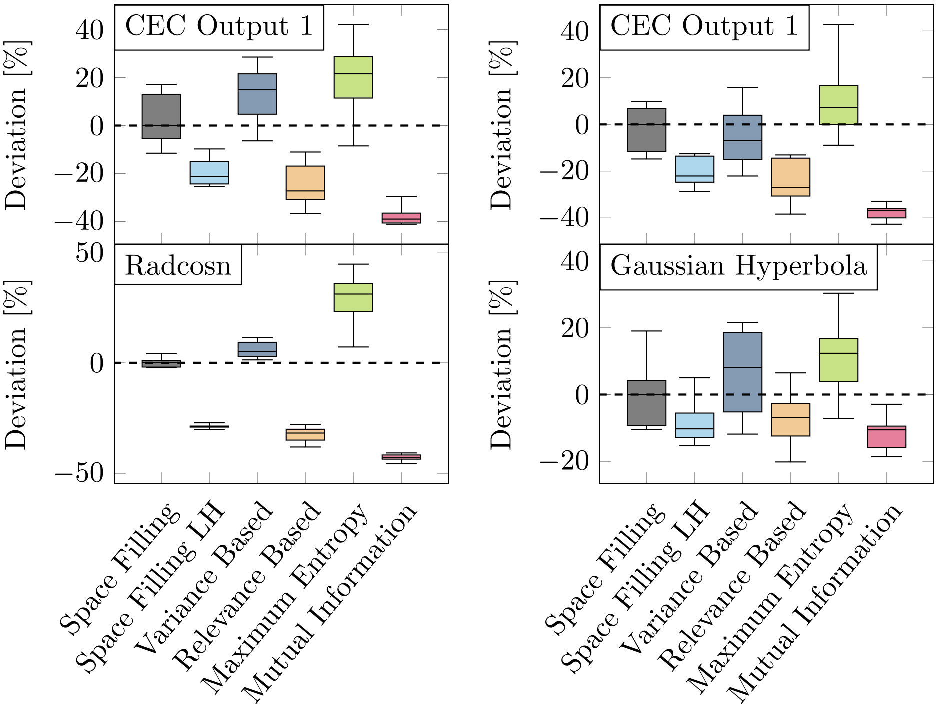


Figure 4.17: Box plots for the AUC value for each test design and each output within the 9-dimensional artificial model simulation. The AUC values show the relative deviation to the median of the space-filling result. A negative value defines a smaller area and therefore improves the validation error compared to the space-filling test design. The whiskers reach from the minimum to the maximum value, which is why no outliers are present.

These designs were compared in a simulation-based environment to a state-of-the-art adaptive maximum entropy design, a non-adaptive space-filling design, and a non-adaptive LH-based space-filling design. The non-adaptive designs were chosen to generally measure the influence of the newly developed designs compared to non-adaptive approaches. The impact was additionally compared to the often applied adaptive maximum entropy design. As simulation models, a diesel engine model and an artificial function model were used.

The two performed simulations showed in general, that the entropy-based MI test design is able to reduce the model error for all outputs during a measurement campaign compared to a space-filling test design. Also the average model error reduction is best compared to all other investigated methods. A ranking in terms of error reduction for the adaptive methods could be identified to be

1. MI test design
2. Relevance-based test design
3. Maximum entropy test design
4. Variance-based test design

with the MI test design as the best method. The relevance-based test design, which uses the Gaussian process length scale parameter as criterion, was comparable but slightly worse than the MI test design. However, the MI test design offers a significantly higher calculation cost. The variance-based test design was not able to generally reduce the model error compared to the space-filling test design. The maximum entropy design performed well within the diesel model simulation but even worse than the space-filling design within the artificial model simulation.

This chapter considered the two open questions

- to find the optimal adaptive test design strategy and
- to handle more than one output.

A solution especially for the optimal adaptive test design strategy was provided. Since both questions are strongly related, there is no simple answer for a best method if more than one output is present. However, solutions were given for the most important test design strategies namely the MI and the relevance-based test design and how a test point is calculated if several outputs are present. The final result to judge the test design performance already incorporated a test design strategy for four different outputs, for which reason both questions were answered within this chapter in a combined simulation.

Test space restrictions were not considered within the search for the best test design. All simulations were performed without any limits in the output domain. The subsequent chapter 5 will find a solution to embed the most promising test design into an environment considering output limits as well.

5 Multicriterial Adaptive Test Design

The main target of this research is to identify an overall strategy for an adaptive test design method that focuses on two criteria. On the one hand, the observed input domain has to be as large as possible, to find a best solution for a subsequent multicriterial optimization task with constraints. In case the measured area does not reflect the optimum, an optimization procedure is not able to solve the cost function appropriately. On the other hand, the model quality of the relevant process outputs needs to be increased as fast as possible in terms of measurement amount. The optimization result significantly benefits from a high model quality to achieve an optimal solution with high certainty.

The last two chapters introduced several methods for both criteria, where a particular best method was identified meeting the objectives. However, these methods where considered individually without any connection. Within this chapter, a combined approach is developed. A strategy is introduced for an overall adaptive test design, dealing with the optimization of both given criteria.

First of all, a combined methodology is developed and incorporated in section 5.1 into an overall test design strategy considering validation points, sorting options and the initial measurement phase. The proposed strategy is tested within a simulation environment for two different test cases in section 5.2. The results are compared regarding model and validation error quality. The calculation duration per test point for the adaptive approach is furthermore evaluated, as it is a crucial criterion for an adaptive procedure running during a test. The last section 5.3 summarizes the findings of the multicriterial adaptive test design.

5.1 Method Combination

The test design criteria taken into consideration are in some ways conflicting. While a boundary search always tries to find measurements at the input domain boundary area to enhance the identified area size, the model quality is increased by a test design focusing high-gradient areas and a generally (potentially distorted) low-discrepancy-based distribution. At each design step during an adaptive procedure, this conflict of objectives has to be solved to find the best compromise. In the following sections, a detailed objective of the combination is given, whereupon a design procedure to combine the most improving strategies with low particular improvement loss is described. The design process is integrated into a non-expert applicable implementation approach, where practical problems like an applied sorting and a data exchange with an automation system are solved.

5.1.1 Objective of the Combination

The main issue of the conflicting targets of a boundary search and model quality focusing test design is described by figure 5.1. A boundary search test design tries to reflect the boundary shape as best as possible. Especially at nonlinear borders many test points need to be placed for an exact boundary description. Test points within the boundary area are not considered, because they do not contribute to the boundary description model. A non-distorted space-filling test design, which is assumed to be an optimal solution to focus model quality in this example, does also reflect the area inside the boundaries. To identify the process behavior, both, boundary measurements and space-filling measurements lying inside the hull, need to be collected. The boundary measurements are only slightly influenced by the boundary shape. A space-filling criterion also takes measurements at linear boundaries because the distance criterion has to be met independent of the boundary shape. This example shows the different test designs if the boundaries are already known. In case they are not, the deviation in the resulting measurement could be even higher.

The combination of both criteria can be very challenging. The objective, however, is strongly affected by the criterion of the overall test. In sequential procedures, the boundaries are investigated in advance of the test design (*2-Stage Offline*). Within an adaptive procedure a sequential identification is possible and applied as well (*2-Stage Full Adaptive*, see e.g. [The+16]). In terms of model quality and measurement duration, a better solution is to recognize boundaries during a test design that generally focuses the model quality (*1-Stage Full Adaptive*). Since the boundary-finding stage has no defined finish within the *1-Stage Full Adaptive* approach, a test outcome

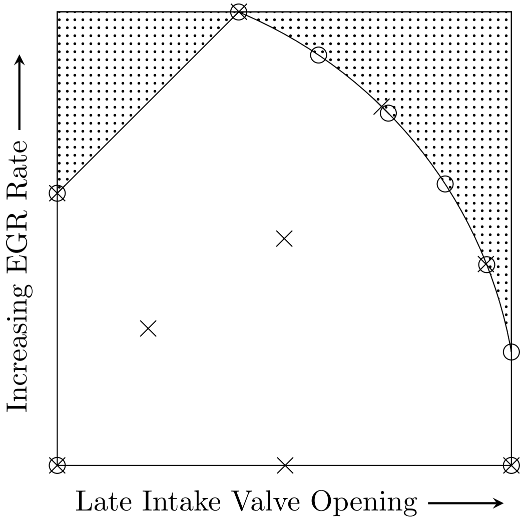


Figure 5.1: Example for different scopes of test designs. Shown is an optimal solution for boundary search (circles) and a space-filling test design (crosses), sequentially created by maximin procedure with a Sobol-based candidate set and starting with the first entry of the candidate set.

target is needed to weight each criterion. Within this research, the identification of the measurable area is considered with higher priority. On the one hand, a high model quality is rated as very important for a good engine simulation model. A very accurate model within a small measured area is not able to predict the global behavior on the other hand. The aim should not be to only select test points that mainly contribute to the boundaries though. A compromise has to be found, which prioritizes the boundary search in case only little information is given.

5.1.2 Test Design Process

The process to find a next test point is divided into two phases. These are not two different stages, but an initial information has to be provided to be able to apply the adaptive test design methods. The following describes the target and implementation of an initial test design calculation. The application of the model-based test design approach in combination with the adaptive test space restriction method within a constrained input domain is given subsequently.

Initial Test Design

An initial test design has to be found that delivers the most possible information for the subsequent adaptive test design. The initial design should be calculable in advance of the test, which means that only the test specification has to be given but no measurement is performed. Within the test specification also input constraints are defined as listed in section 2.3.4. Since these are discretized by a Sobol candidate set, the initial test design utilizes the same data base for the test point search.

Commonly no information about the process behavior is given in advance, which is why a simple low-discrepancy design could be applied. However, during the beginning of a measurement campaign the main target should be to identify the operable area. The objective is an initial test design that maximizes the measured input domain. The number of test points should be at least $\mathrm{dim} + 1$ for dim inputs to be able to calculate a convex hull and to determine the exclusion area B as investigated in chapter 3. A slightly higher amount of initial, boundary searching measurements is preferred for measured area increase and robustification reasons. These measurements mainly contribute to the boundary search and are crucial for the quality of the subsequent adaptive design and especially the exclusion area size.

A candidate exchange algorithm is introduced to identify an initial test design, which maximizes the convex hull volume. A brute-force approach could be applied as well, but takes a very high amount of calculations. As an example, a selection of 10 test points from 20 candidates takes 184756 possible solutions, which increases exponentially with the amount of candidates. A solution for 30000 candidates, as used during these investigations, is therefore unreasonable. An amount of $2 \cdot \mathrm{dim}$ candidates is selected by maximin in the first step. The convex hull of these candidates, calculated by the qhull method [Bar19], provides the volume and enables the algorithm to identify if test points lie inside the hull or not. At first the points lying inside the hull are substituted by candidates, which are assumed to increase the hull volume greatly. These are identified by an orthogonal distance criterion to the hull-generating hyperplanes, where the particular candidate with highest distance is chosen. This criterion follows the qhull algorithm, where it is used to incorporate a largest possible area as well. Once all candidates are convex hull-defining vertices, the candidate exchange procedure is launched. In each iteration, the furthest candidate

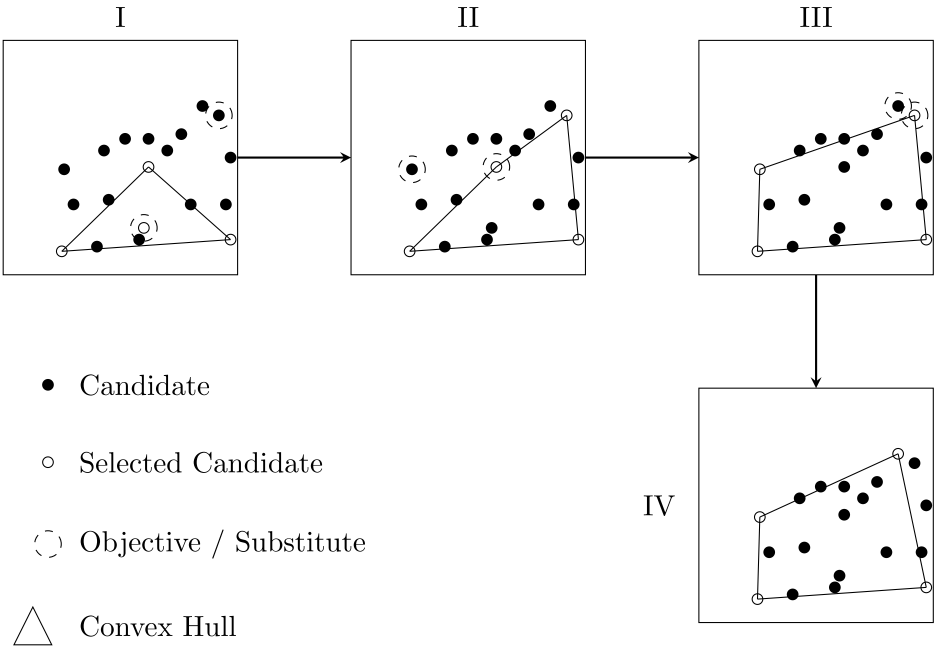


Figure 5.2: Initial test plan calculation strategy. Given a candidate set, a test design with 4 test points is the objective. Therefore 4 initial test points are selected (I) and points inside the hull are substituted by the hull's furthest candidates (II). Subsequently, hull points are exchanged by furthest candidates until there is no more hull volume increase present (III and IV).

again is identified by the highest orthogonal distance to each hull hyperplane. The nearest hull point, defined by the Euclidean distance to the furthest candidate, is substituted by the considered candidate and the hull is calculated again. This exchange is executed until the hull volume does not increase anymore or a maximum amount of iterations is reached. The algorithm is schematically shown for two inputs in figure 5.2. In this case the procedure terminates after 4 steps, because the hull volume does not increase further.

Criteria Combined Adaptive Design

Several possible solutions exist for a combination of boundary search and selective model quality optimization. Discussed solutions are

- a configurable switching of targets,
- a model quality focus considering boundary prediction and
- a target compromise.

The configurable switching of targets, which corresponds to a *2-Stage Full Adaptive* approach, offers a necessary switching requirement as main disadvantage. Once the process boundaries are identified to an acceptable quality, the test design switches to a model quality focus. Therefore, the boundary identification measurements are designed to mainly contribute to the operable area definition, where their influence on model quality is not considered. As the requirements for a sufficient boundary quality are hard to guess for all use cases, the switch criterion has to be user-configurable. This introduced flexibility forces the user to make a decision and contradicts the requirement for an easy to apply overall process.

A method that solely focuses on model quality is able to significantly improve the average model quality, as shown in chapter 4. An improvement is achieved if the boundaries are known well and the test design is adjusted accordingly. However, the result about the measured input domain volume is random and not controlled by the test design method. Additionally, boundaries are mostly not known beforehand. For these reasons, this approach is inappropriate as well.

A compromise between model quality focus and boundary investigation is the most suitable procedure. Such a method fulfills the requirement about a continuous input domain volume increase as well as a model quality focus. However, the application of a compromising method could end up in a parametric functionality, which is not the target. The strategy introduced here follows the two fundamentals:

1. Design space-filling in unknown areas
2. Design model-based in known areas

The area selection for the next test point is defined by the space-filling criterion in the non-distorted input domain. That is, a next test point is firstly planned by the maximin criterion. In case the test point is not part of the convex hull of already taken measurements and not part of the exclusion area B, it is used as next measurement as it is. Only in case it is part of the convex hull, the designed point is refused to find another one

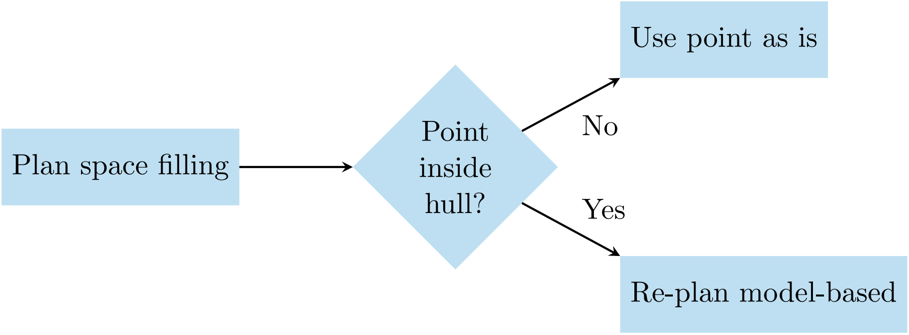


Figure 5.3: Decision process for the utilized test design method

that maximally contributes to the model quality within the measured area. This general procedure is shown in figure 5.3.

The detailed process differs for lower and higher dimensions. For lower input dimensions, a straightforward procedure is applicable. Since the convex hull hyperplanes are calculable with comparatively low effort, they are present in each test design iteration. For that reason, the hyperplane-based exclusion strategy is applied (see section 3.3.1), where it is affordable to erase all candidates within the exclusion area B from the candidate set $\boldsymbol{S}$ prior to the main test point search. A next test point is generated by applying the maximin criterion to the reduced candidate set $\boldsymbol{S}_r$, given the already taken measurements $\boldsymbol{X}$

$$\boldsymbol{x}'^{*} = \underset{\boldsymbol{S}_r}{\operatorname{argmax}} \min_{\boldsymbol{x} \in \boldsymbol{X}} d(\boldsymbol{x}, \boldsymbol{S}_r). \tag{5.1}$$

which results in a potential next test point $\boldsymbol{x}'^{*}$. In case this test point is not part of the convex hull of the measurements $\mathrm{co}(\boldsymbol{X})$, it is directly used as next test point. A possible solution would be to find a test point that maximally enlarges the convex hull if measurable. However, a space-filling test point is used, which is a compromise between hull enlargement and model quality improvement. As the design criterion of a space-filling test point maximizes the distance to all measurements, it provides a test point with high Euclidean distance to the measured hull and therefore still contributes to the hull enlargement strongly. In case it is part of the convex hull, this test point is refused to find a more appropriate solution by the model-based test point search. A model quality focused test point

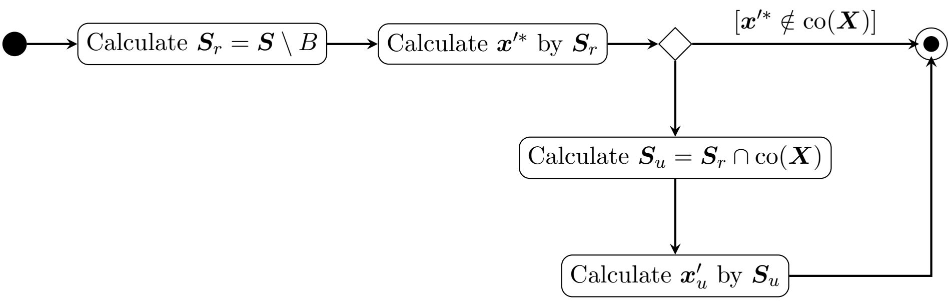

Calculate $\boldsymbol{S}_r = \boldsymbol{S} \setminus B$
Calculate $\boldsymbol{x}'^{*}$ by $\boldsymbol{S}_r$
$[\boldsymbol{x}'^{*} \notin \mathrm{co}(\boldsymbol{X})]$
Calculate $\boldsymbol{S}_u = \boldsymbol{S}_r \cap \mathrm{co}(\boldsymbol{X})$
Calculate $\boldsymbol{x}'_u$ by $\boldsymbol{S}_u$

Figure 5.4: Test point calculation by combined test planning criteria and application of the hyperplane-based exclusion area check

is calculated with the candidate set $\boldsymbol{S}_r$ being reduced to a set with only candidates inside the convex hull $\boldsymbol{S}_u$. A next test point is derived from the reduced set by the particular model-based test design approach. The whole process for the hyperplane-based exclusion procedure is given in figure 5.4.

Figure 5.5 shows the process for high input dimensions. This procedure takes place in case the convex hull hyperplanes are not calculated, i.e. for dimensions 7 and 8 in case many test points have to be evaluated for being part of the convex hull and at dimensions higher than 8 (refer to section 3.3.2 for detailed explanation). A main difference to the procedure for low input dimensions is the iterative approach in both main steps, selecting a valid candidate by maximin as well as by the model-based method. However, the crucial decision if a next test point contributes to model quality or measured input domain volume is made in the very beginning. A potential next test point $\boldsymbol{x}'^{*}$ is designed by maximin to find a space-filling test point given already taken measurements $\boldsymbol{X}$ and the full candidate set $\boldsymbol{S}$. In case this test point is part of the convex hull of the measurements $\mathrm{co}(\boldsymbol{X})$ the subsequent measurement will not increase the measured area. Therefore, $\boldsymbol{x}'^{*}$ is rejected and a next test point is designed by the model-based design strategy (figure 5.5 lower path). Since it would be a high effort to only consider candidates within the convex hull of measurements, a derived potential solution contributing only to model quality $\boldsymbol{x}'_u$ is tested for being part of the convex hull. If it is not, the candidate is temporarily erased and the next optimal point is calculated. In case it is part of the convex hull it is used as next test point. Especially if the convex set solution by

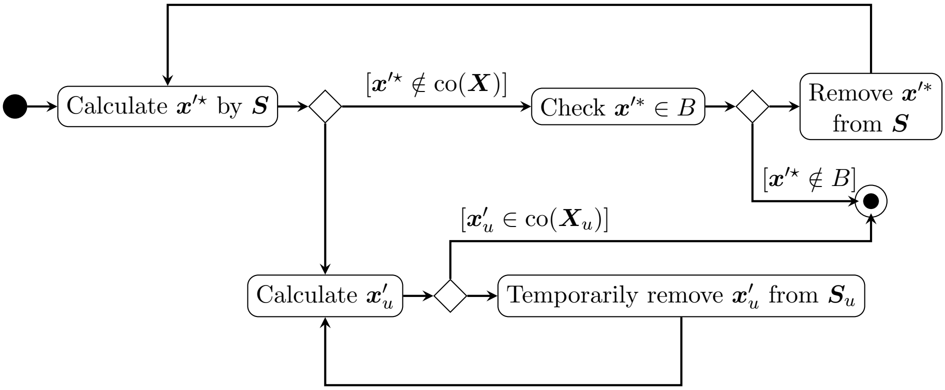

Calculate $\boldsymbol{x}'^{\star}$ by $\boldsymbol{S}$
$[\boldsymbol{x}'^{\star} \notin \mathrm{co}(\boldsymbol{X})]$
Check $\boldsymbol{x}'^{*} \in B$
Remove $\boldsymbol{x}'^{*}$ from $\boldsymbol{S}$
$[\boldsymbol{x}'^{\star} \notin B]$
$[\boldsymbol{x}'_u \in \mathrm{co}(\boldsymbol{X}_u)]$
Calculate $\boldsymbol{x}'_u$
Temporarily remove $\boldsymbol{x}'_u$ from $\boldsymbol{S}_u$

Figure 5.5: Test point calculation by combined test planning criteria and application of the iterative exclusion area check

the GJK method is applied, the evaluation cost is directly correlated with the amount of evaluation points and therefore this iterative approach is necessary.

Within the upper path of figure 5.5, the procedure is shown for the hull enlargement strategy. The initially calculated space-filling test point $\boldsymbol{x}'^{*}$ is not part of the measured area $\mathrm{co}(\boldsymbol{X})$, which drives the decision to perform a boundary search measurement. The derived test point is checked for the exclusion area B by the convex cone strategy introduced in section 3.3.2. Only if the selection is part of any local exclusion area, the candidate is ultimately removed from the candidate set $\boldsymbol{S}$ and the whole algorithm starts over again. Otherwise the space-filling test point is taken as next measurement target.

The described structure offers the possibility to implement any type of design strategy for the model improving path. A rating of different strategies is given in chapter 4 with the MI design as the best overall performing. Compared to the relevance-based design, this test design offers a higher calculation duration if the design considers only one output. In case the test design focuses several outputs, the calculation duration increases multiplicatively with the amount of outputs. Within the multicriterial test design, the calculation duration is crucial, because several different calculation steps could be necessary. Especially within the high dimension approach, a strategy with low calculation duration should be favored. Therefore, the second best rated strategy, the relevance-based test design, is applied to the multicriterial design. This design requires roughly the same calculation amount as a space-filling test design. Additionally, due to the combined calculation for several outputs, the duration does not increase with the amount of outputs different than with the MI test design.

5.1.3 Overall System Integration

The outcome of the investigations has to be applicable to engine tests at an engine test bench. Several restrictions and requirements have to be fulfilled for running the adaptive test design within an MBC process. The very first requirement is the utilization of an interface to existing test bench automation systems. The designed algorithms should not only work within a designed example measurement but have to serve as a methodological base for different MBC tasks. A lean interface is described and integrated to the software IAV Kasai. An introduction to the software is given in appendix E.

Another requirement is the investigation of a test error. Additional test points have to be planned, which strategy is different in an adaptive test design compared to a non-adaptive one. A user-given sorting of test points is an additional challenge for the adaptive test design and is described in the following.

Interface Definition

The introduction of an adaptive test design process is only successful, if the developed system is applicable to the engineer's tool chain with low effort. The main requirement to the interface of the design method is the applicability to existing automation systems. Therefore, a well established interface is preferable to be able to implement it into every existing system. The adaptive test design is integrated into IAV Kasai, which already provides all necessary test definition steps. A test is defined by the input parameters, their ranges, and the different constraints as discussed in section 2.3.4. The result is a Sobol-based candidate set, fulfilling all user-designed constraints. This set is the base for the test design process as introduced in the preceding section.

Given the candidate set and the algorithms implemented in the software, an interface is needed for the communication with the automation system to provide the newly designed test point and to receive the neces-

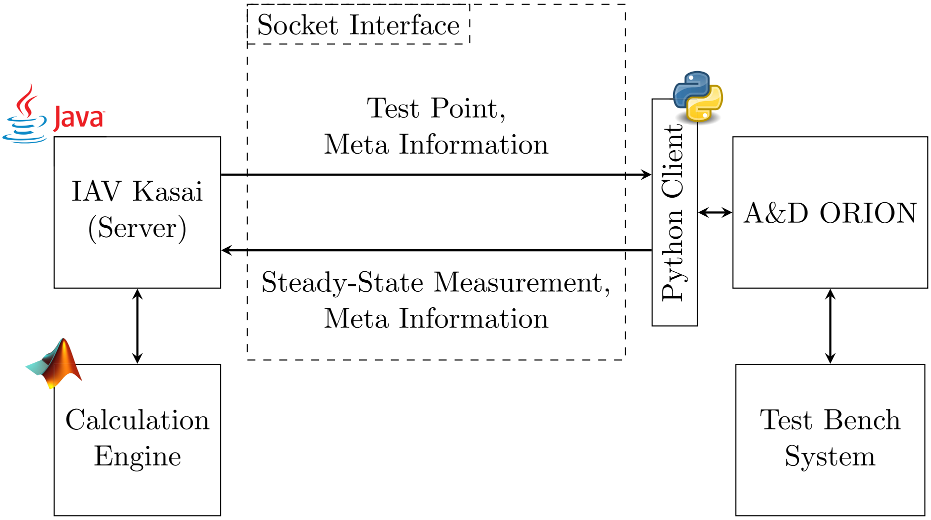


Figure 5.6: Interface configuration for IAV Kasai with example implementation for A&D ORION and interposed Python-based client. The calculation engine is in charge for the execution of the developed algorithms.

sary measurement data to train the models. A transport control protocol (TCP)-based socket interface provides a suitable architecture and is available in all established programming languages. With this interface type, it is possible to run the algorithms and the automation system on the same computer or on different computers with communication via network. Fundamentally, IAV Kasai opens a socket server, addressable for a client that is implemented on the automation system's side. This configuration enables the automation system to ask for a next test point where the answer is the next setting to measure. Once the measurement of this setting is done, the mean measurement result is sent to the server, where it is processed and stored in the database. The interface definition consists of several additional meta information in both calls, where at least an information about the boundary status of a measurement must be provided by the automation system. The complete communication architecture is given in figure 5.6. The left side implements all methods as developed in this research, whereas the right side is exchangeable but implemented in the shown way.

Validation Test Design

In addition to measurements used to train the GP models, validation measurements are commonly performed to rate the generalization error of a model. The main target of a validation point set is to access the mean generalization error of a given model within the input domain. The more validation measurements are present, the better is the estimation of the generalization error. However, since the validation measurements are not used to train the model, each of these measurements does not contribute to a model quality increase. Hence, a space-filling design is suitable to keep the amount of measurements at a low level for a good trade-off between amount of measurements and test quality.

Contrary to a non-adaptive approach, an adaptive test plan calculation provides the possibility to calculate a validation point setting that is measurable in terms of satisfying the boundary constraints. At each test stage, the convex area of the measured training points gives rise about the measurable and also interesting region within the input domain. This scope is utilized to plan a validation point. Space filling in the sense of validation measurements is only related to already taken validation measurements. A next space-filling point utilizes the candidate set $\boldsymbol{S}_v$ to design a test point by maximin

$$\boldsymbol{x}_v = \underset{\boldsymbol{S}_v}{\operatorname{argmax}} \min_{\boldsymbol{x}_v \in \boldsymbol{X}_v} d(\boldsymbol{x}_v, \boldsymbol{S}_v). \tag{5.2}$$

where $\boldsymbol{S}_v$ is given by the intersection of the original candidate set $\boldsymbol{S}$, reduced by all model training points $\boldsymbol{X}$ and all candidates not within the already measured convex hull region

$$\boldsymbol{S}_v = (\boldsymbol{S} \setminus \boldsymbol{X}) \cap \operatorname{co}(\boldsymbol{X}). \tag{5.3}$$

The amount of validation points is defined relative to the amount of measurements. That is, they are iteratively planned during the measurement campaign to always fulfill a user-defined relative amount of validation points. Irrespective of the stage at which the process is interrupted, the demanded amount of validation measurements is present. With a growing convex hull of training data, the generalization error is judged within this growing area by means of the iterative design approach.

Test Point Sorting

Different to a non-adaptive test design, a calibration engineer-defined sorting option is a critical issue in an adaptive approach. A very common option is a meander sorting of engine speed and load. These two parameters, commonly indicating a measurement procedure as global in contrast to a local identification without varying speed and load, have a high influence on most engine outputs. Temperatures, in particular, are highly affected by load but also by speed. A frequent change of these parameters entails a high effort for the stabilization of temperatures and is reduced by a meander-based sorting.

In case all tests are known in advance, a sorting could be applied easily. However, with an iterative approach with only one test point planned in advance of a measurement, a sorting is not applicable in the usual sense. Two different procedures are proposed to obtain a pseudo-sorting. The first solution is to plan a block of test points with the adaptive approach iteratively. All tests in the reverse direction of the last measurement are erased from the block and the points left over are sorted by the applied sorting option. This procedure is exemplarily shown in figure 5.7 for an increasing speed and load meander sorting option. Once the meander is fully passed and no more test points remain, the full domain is permitted for a test point search again and the next measurement is the first after sorting the block. Two negative properties are present for this approach. Firstly, a high number of necessary test points leads to a high calculation duration for a next test point. The introduced methods exhibit a high computing time, especially for high-dimensional tests, which is why this approach seems to be non-constructive. Secondly, the valid test points are calculated with the assumption that the erased test points are going to be measured. The test design quality could suffers because the next test point is likely not the next most informative measurement.

Another approach comprises the sorting option during the test point search. The domain for the test point search is simply reduced to incorporate a relatively small scope. That is, starting at the last measurement, for example only the next 10 % of the full domain fulfilling the sorting criterion is approved for a test point search. Since a candidate set is used for the test design, all candidates not fulfilling this criterion are temporarily erased. Once the remaining scope does not exhibit the defined width, it is extended by the necessary width starting at the origin. The transition from

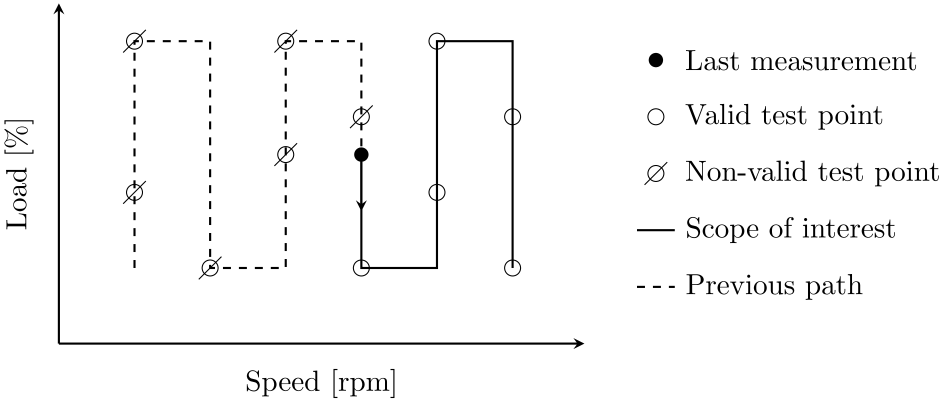


Figure 5.7: Procedure to erase test points that do not fulfill a meander sorting option for speed and load. The last measurement is decisive to distinguish between test points in the wrong and in the right direction. An increasing speed and a meander sorting of load with initial increase is given for this example.

the end to the beginning is ensured by this procedure. Due to the reduced amount of candidates valid for the test design and the fact that only one test point is necessary instead of a block, the test point search duration is decreased significantly. However, the maximum potential of the test design methods is not obtained, because the temporary scope of interest is smaller than the test scope. A method without losing any potential is in conflict with the sorting option though. The higher the width for the test planning scope is chosen, the higher the test design quality is. At the same time, the sorting criterion is fulfilled with lower quality. Since this method chooses a next test point with respect to the test design criteria and the calculation duration is significantly lower to a block-based test design, it is chosen with a scope width of 10 % for this research as a compromise between sorting quality and test information content. This value could be parameterizable to let the calibration engineer define the importance of his or her applied sorting.

Calculation Scheduling

In section 2.5 and with figure 2.13, the permitted duration for a next test point search is defined to be 30 seconds. A higher duration is present between two test point requests, which is defined to be 300 seconds. To reduce the probability for the measurement procedure to wait for a test point calculation, at least one maybe non-optimal test point has to be present. After the initial test design is completely measured, two test points are calculated and only the first is provided to be measured. Once the measurement is present, a model training is executed. During the subsequent iteration the second test point is provided, where another test point is calculated in parallel with the existing models. Since this test point is determined without the measurement of the second test point, it will not provide the maximum information content. However, only marginal loss is present but the design algorithm has at maximum the full iteration time of 300 seconds for a test point calculation available. The scheduling of calculation tasks is schematically shown in figure 5.8. The main target is to always provide a next test point without interruption at the beginning of a set parameter action. Even if the model training calculation duration is longer than the parameter reset takes, the test design is executed without consideration of any other tasks. Therefore the model training is interrupted until the test design completes. Since it can occur that some outputs are not up to date during the test design calculation, the test point search could slightly suffer additionally. However, the primary target is to not interrupt the measurement procedure.

The influence of all given procedures, namely the test design strategy, the applied sorting procedure, and the test point calculation scheduling, on the test design quality will be gathered in the subsequent section.

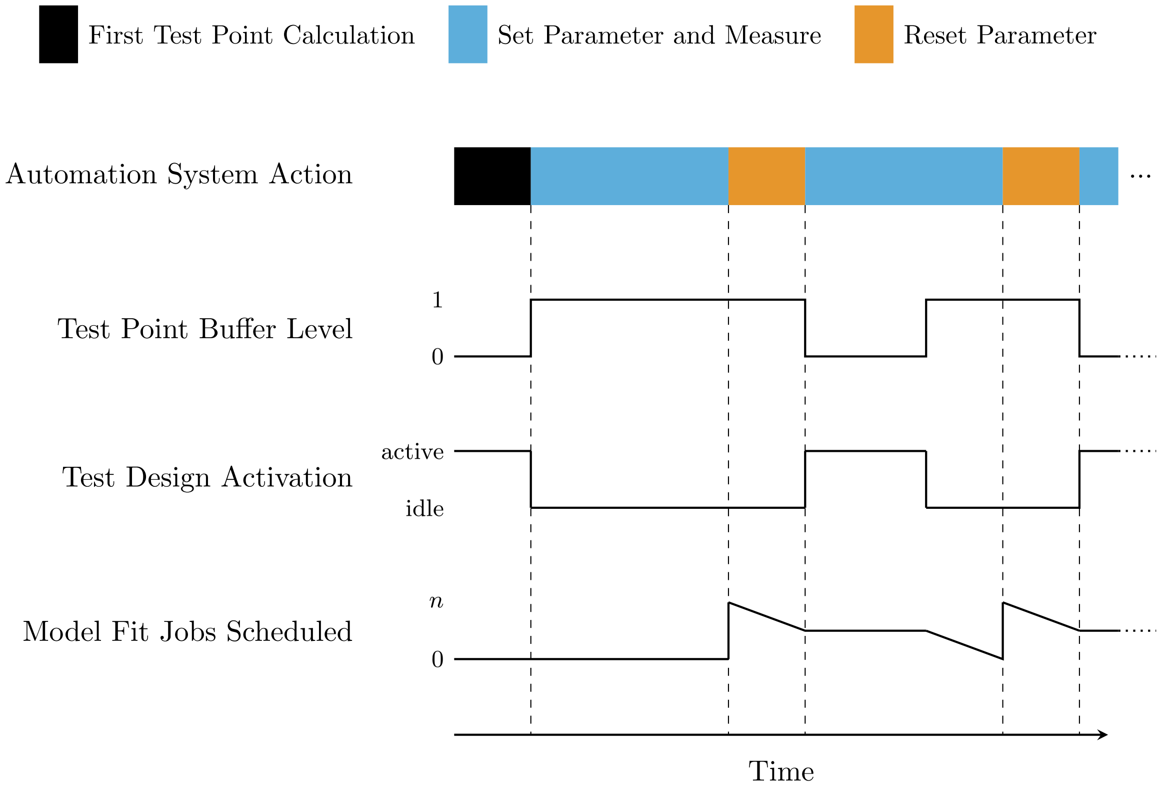


Figure 5.8: Scheduling of the different calculation tasks during the measurement procedure. The tasks are triggered by the automation system process queries. The model fit tasks for the n outputs are scheduled at lower priority and are interrupted by the test design. The highly prioritized test design task increases the test point buffer level to ensure a valid test point at the beginning of each set parameter and measurement phase.

5.2 Simulation-Based Investigation

A simulation is performed to judge the effect of the given multicriterial design strategy. The test design algorithms in their base do not differ from those introduced in chapter 3 and chapter 4. However, the exclusion area method was only tested with a space-filling design and the model-based design was not tested within a restricted input domain. Both performed very well within their scope, but a final test has to show their combined benefit by the introduced multicriterial test design. Additionally, the given solutions to the necessary test execution requirements have an influence that cannot be neglected and that needs to be considered in the overall context. Therefore, the simulation environment is introduced in the following and the test results are discussed.

5.2.1 Simulation Environment

The multicriterial structure is implemented to IAV Kasai, which is why the interface description of figure 5.6 takes place within the simulation. The iterative exclusion procedure is applied at input dimensions higher than 6, the hyperplane-based exclusion area check is executed otherwise. Since a simulation has far lower requirements to the test execution compared to a real-time engine test, an automation system is not necessary for the simulation procedure. Therefore, the interface to IAV Kasai is implemented in a MathWorks® MATLAB® environment, in which the simulation models are present. The same models, including the noise model, as discussed in chapter 4 are utilized for the simulation-based investigation. The input noise is set to $\sigma_i = 0.02$, the output noise to $\sigma_o = 0.01$ and the process noise to $\sigma_w = 0.0005$, which is equal to the preceding simulations. Since a global test is examined for both models but the number of inputs has to stay the same, the two post-injection parameters timing and mass (see figure 3.10) are not varied for the diesel engine model. The inputs for the artificial model stay the same, where the first two inputs are considered as speed and load. The test point sorting is configured to be ascending within speed and load with a meander sorting over load. A validation point share of 10 % in relation to the training points has to be the outcome. The adaptive design is compared to a non-adaptive space-filling test design that is calculated by IAV Kasai and is sorted in advance of the measurement. The non-adaptive test plan is divided into five blocks, where each fulfills the space-filling criterion and the sorting option. Therefore, the test runs five times through the whole speed and load domain.

The objective of the simulation is to compare both, the average model quality as well as the measured input domain size. A constrained input domain is necessary to judge the effectiveness of the combined approach. Within the 7-dimensional diesel model the convex hull of the model data is used as restriction. The 9-dimensional artificial model utilizes the same procedure as described in section 3.4.2 but with a slightly modified lower bound. This setting leads to an approximate restriction of 47 % of the input domain, meaning the measurable area captures 53 % of the hypercubic input domain. The restrictions are given in table 5.1. Each simulation setting is run 10 times to allow a statistically confirmed evaluation of the results and to rate the fluctuation between each run for both methods. A total amount of 905 training points and 90 validation points are simulated for both methods in each run.

Table 5.1: Boundary constraints definition

Model Type	Dimension	Lower Bound	Upper Bound	Further Restriction
Artificial	9	0, 0, 1, 0.1	-	CEC2009 Constraints
Diesel Engine	7	-	-	Convex Hull

5.2.2 Discussion

The discussion of the simulation outcome starts with the evaluation of the most important model quality outcome. Additionally, a comparison between the validation point designs is given and an assessment of the calculation duration for the adaptive approach is carried out.

Model Quality

To rate the model error reduction trend over a simulation campaign, the AUC value is being considered for each output solely. A visualization of the result values and their fluctuation for the 7-dimensional diesel engine simulation is given by the box plots in figure 5.9. Since for all values the normed deviation to the median of the particular non-adaptive result is shown, an improvement is identifiable by a negative deviation. However, the fluctuation, especially the box width, still has to be compared. Both the average area under the validation error curve as well as the variation between each simulation run is strongly reduced by the adaptive multicriterial method, shown in the upper right diagram. Even the worst run of each output offers an improvement compared to the median value of the non-adaptive test design shown by the upper whisker is not exceeding 0 %. The model quality change for the soot model shows the least improvement, but in comparison to the model-based investigation results in chapter 4, the change in quality is significantly better. A rate of about 11 % of the training points are planned within the hull, which is by means of the relevance-based test design. This design method is applied at first after 630 fit measurements, because a higher priority is put on the input domain exposure. To demonstrate the difference to a solely model-based design, the result for a simulation without considering boundaries is shown as adaptive model-based design in figure 5.9 in the lower plot. The differences in average and fluctuation outcome for the soot and CO model are considerable with even a deterioration for the soot model compared to the non-adaptive approach. The fuel mass and temperature before turbine model is still improved but the quantity is clearly lower compared to the multicriterial approach.

A similar beneficial result is given by comparing the convex hull volume increase. The mean volume trend for both methods is drawn in figure 5.10. The difference in the very beginning is on the one hand influenced by the initial test design that is designed for a maximum volume increase within the multicriterial design. On the other hand, the non-adaptive approach has to measure 190 training test points to run through the whole speed and load domain conditioned by the block-based sorting. The adaptive test design forces more operating point changes than the non-adaptive approach in this example. However, with 190 measurements the adaptive approach still offers a significantly higher volume, which then starts to increase further due to the erasure of immeasurable candidates. In case less operating point changes should be allowed, the candidate scope width could be reduced, forcing the adaptive algorithms to find a next test point closer to the current operating point. A very low scope width is yet not recommended, because a strong restriction of candidates to choose from lowers the test

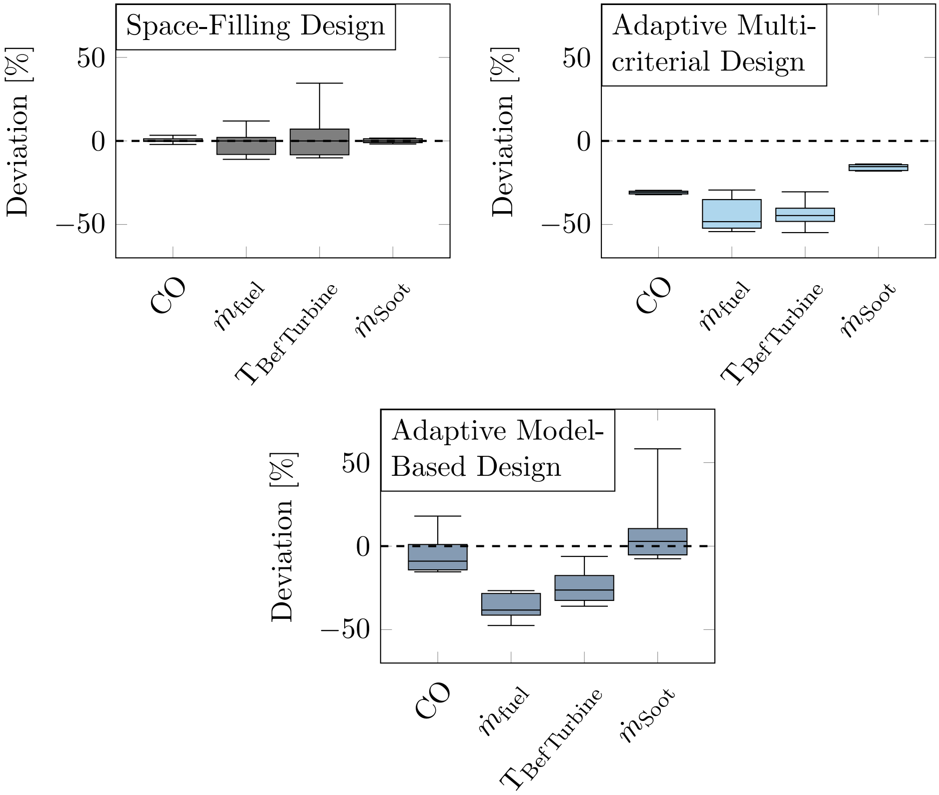


Figure 5.9: Result of the 7-dimensional global diesel engine simulation for both test design methods (top) and for comparison for a solely model-based test design simulation (bottom) with each 905 training measurements. The boxes show the fluctuation for the AUC value between the 10 simulation runs with minimum and maximum value given by the whiskers. All values show the relative deviation to the median AUC value of the particular space-filling design result.

design quality. However, with this configuration a significant benefit in hull volume increase rate could be achieved. Starting from 190 measurements the non-adaptive-based hull volume increase slows down, whereas the adaptive-based hull volume increase starts to gain further. With 905 training measurements the convex hull is more than twice as large for the adaptive design, meaning the measured area is more than doubled compared to the non-adaptive design. To complete the comparison to an adaptive, solely model-based design, the hull volume result is shown as well. A hull volume that is even lower than the space-filling design is present starting from about 140 measurements. This trend originates from the distortion of the input domain. The space-filling criterion is applied in the distorted domain, whereas the hull volume is calculated in the original domain. Due to the distorted design, the distance criterion in some input directions is prioritized lower but their impact on the hull volume is high, leading to the reduced hull volume increase. This approach therefore would lead to a very restricted optimization in case the convex hull or any criterion based on the measured domain is used as an optimization constraint.

Comparing the results of the 9-dimensional artificial model simulation in figure 5.11, the benefits of the adaptive approach are even higher compared to the space-filling design. A reduction down to 40 % in arithmetic mean for the AUC value is present for CEC output 2 with a lowest relative AUC in a single run down to 24 % compared to the space-filling design median result. These values are a huge advantage for the adaptive approach, making it even better that all other outputs have been optimized as well. The radcosn function shows the smallest improvement with still 15.5 % improvement in arithmetic mean and median. A single run offered a deterioration for the Gaussian hyperbola output compared to the non-adaptive median result by 20 %, showing the necessity of repetitions in this test case to rate the average improvement. The fluctuation between the simulation runs changes slightly with an insignificant reduction in quantile size for the adaptive approach. However, this still is a positive result, because an adaptive procedure has to be robust against noisy measurements. A non-adaptive test design does not react to noise, where an adaptive design could misinterpret noisy measurements and reduce the test design quality, but this is not the case with the multicriterial design.

The convex hull volume increase for the 9-dimensional simulation also highlights a benefit for the adaptive approach during the whole simula-

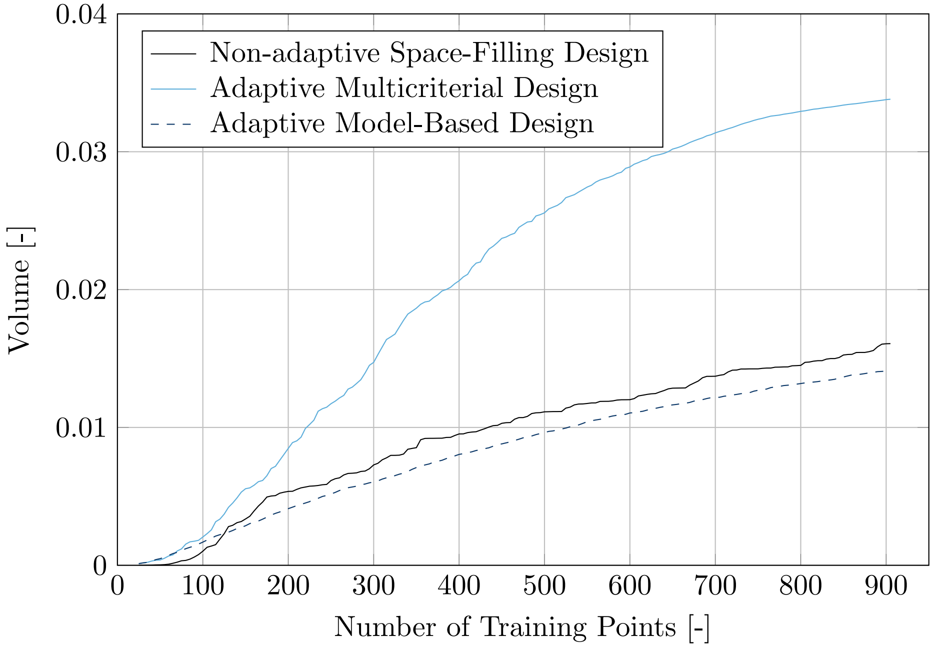


Figure 5.10: Convex hull volume increase for both, the non-adaptive space-filling design as well as the adaptive multicriterial design averaged over 10 simulation runs within the 7-dimensional global diesel engine simulation. For comparison, the result of a simulation with a solely model-based test design is shown dashed.

tion. The block-based sorting at the non-adaptive simulation is apparent in the hull increase in figure 5.12 by the sharp kink after each 190 training points. These kinks are the result of measurements in the different input regions by the sorting of the first two inputs. A more smooth volume increase is present for the adaptive approach due to a more often operating point change as discussed with the 7-dimensional simulation result. The convex hull size results in a 20 % higher volume for the adaptive approach. This smaller improvement compared to the 7-dimensional case is a result of both, a higher-dimensional test case and less restrictions of the input domain by simulated boundaries. However, both the hull volume size and all model qualities are significantly improved by the adaptive multicriterial test design.

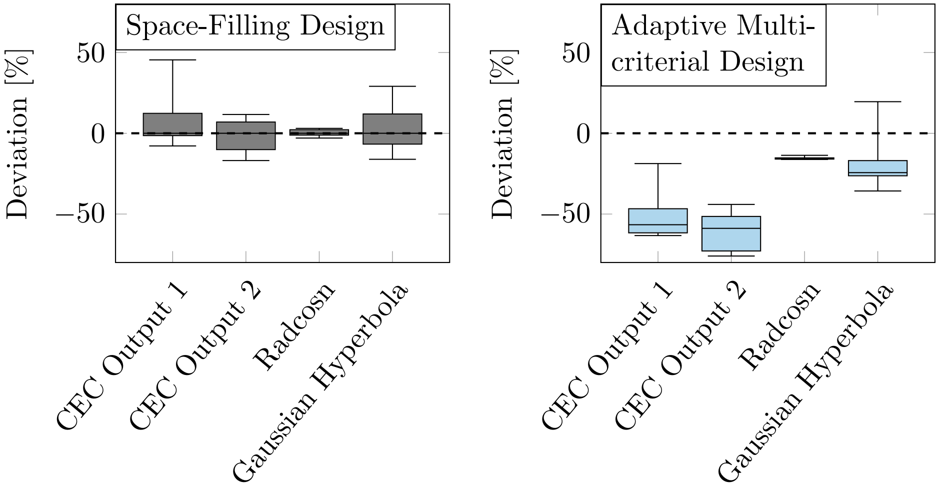


Figure 5.11: Result of the 9-dimensional global artificial model simulation for both test design methods with 905 measurements. The boxes show the fluctuation for the AUC value between the 10 simulation runs with minimum and maximum value given by the whiskers. All values show the relative deviation to the median AUC value of the particular space-filling design result.

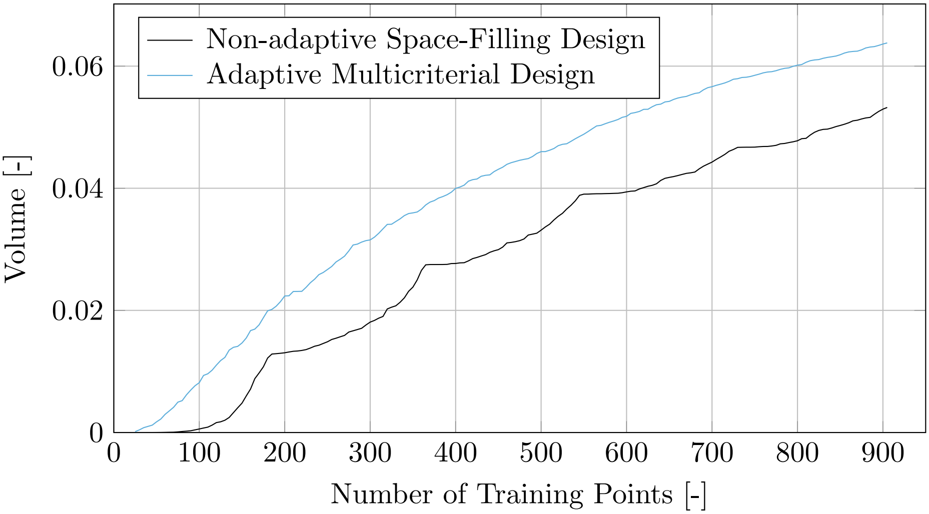


Figure 5.12: Convex hull volume increase for both, the non-adaptive space-filling design and the adaptive multicriterial design averaged over 10 simulation runs within the 9-dimensional global artificial model simulation

Validation Design

The objective of a validation error is to achieve an identification quantity for the prediction capability of a trained model. The validation measurements are not considered during model training but only used to rate the deviation between the model prediction and the measured process outcome. These measurements originate from the same process with the same input, process, and output noise, which is why the deviation between model prediction and validation measurement is never zero. However, the validation test point location in the input domain is crucial regarding the validity of the overall validation error rating. For both simulations, the validation error is rated to compare the adaptive validation test point planning within the measured area with a non-adaptive space-filling validation point design in the hyper-cubic input domain. Since the error within the measured area at each stage of the measurement is most important, a true validation RMSE r_t is calculated by consulting only those candidates of a set with 50000 Sobol-based candidates lying inside the convex hull. In steps of five training points, the error r_t is compared to the validation RMSE given the planned validation points r_p by calculating the RMSE of both trends

$$r = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_{t,i} - r_{p,i})^2} \tag{5.4}$$

to define a quality criterion for the accuracy of the validation error by using the planned validation points and given n different count of training points. The error trend RMSE is calculated for both test design simulation results, averaged over simulation runs and set in relation. To simply recognize an improvement or deterioration of the adaptively planned validation point design compared to the non-adaptive design a relative deviation is calculated by

$$r_{rel} = 100\left(\frac{r_{Adaptive}}{r_{NonAdaptive}} - 1\right). \tag{5.5}$$

The results for both simulations and each output are given in figure 5.13. A very diverse result is the outcome of the validation error comparison. While the validation error of the adaptive design performs better within the 7-dimensional diesel engine simulation for 3 outputs up to half of the non-adaptive error, it shows a significantly worse error for the soot mass

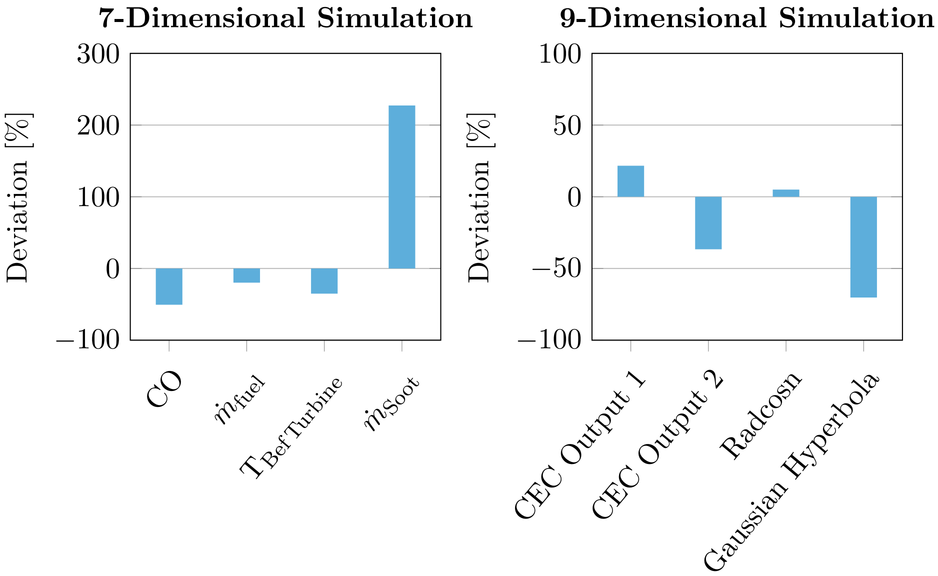


Figure 5.13: Comparison of the validation point design quality. An RMSE of the validation RMSE given the planned validation measurements and a large low-discrepancy test set fulfilling the convex hull constraint is calculated. For each RMSE of the adaptive multicriterial result a relative deviation to the result of the non-adaptive space-filling result is provided. A relative deviation is given for each output of the 7-dimensional diesel engine simulation (left) and of the 9-dimensional artificial model simulation (right).

emission. The high deviation for the soot mass error is mainly explained by the different size of the measured areas. Due to the large measurement area with the adaptive approach and the tendency of soot emissions to exhibit high gradients and therefore high values near engine boundary areas, higher measurement values are taken with the adaptive approach. Within the non-adaptive approach these areas are not covered by the reference validation error, which is calculated by candidates only being part of the measured convex hull. Therefore the reference validation error shows a high deviation between adaptive and non-adaptive approach. A second reason for the high deviation is that the adaptively planned validation points tend to overweight the center of the measured area in contrast to the in advance planned validation points, which resolve the boundary area more accurate. Therefore the validation error is underestimated within the adaptive approach for processes with high gradients near the boundary area in case the input domain is strongly restricted.

Up to a third of the error for the non-adaptive approach is achieved within the 9-dimensional artificial model simulation for the Gaussian hyperbola function. An error quality improvement is achieved on average for the 9-dimensional simulation, with a slight deterioration for two outputs but a strong improvement for the other two outputs. There are less boundaries present within this test case, leading to a good overall result for the adaptively planned validation points.

Calculation Duration

Attention has to be paid to the calculation duration per test point. A requirement for the maximum allowed duration was defined to be 300 seconds to avoid undesired waiting times during a test bench measurement. Figure 5.14 (left) gives an overview over the calculation duration per test point within the 7-dimensional simulation. At almost each test point count a duration lower than 300 seconds is present during the simulation. Unfortunately, 7 test points exhibit a much higher duration with up to 1350 seconds. The high effort in calculation results from the strongly restricted input domain, providing only 7.6 % operable area, that forces the iterative exclusion procedure to erase several candidates in the estimated non operable region. Since this calculation is time consuming, such a high duration appears. Once the most candidates within these regions are erased, the duration only increases slightly with the amount of training points. This is similar for the 9-dimensional simulation, which does not need to erase as many candidates, because the restricted area is smaller. Each test point is calculated much faster than the provided requirement allows in this test case. Different solutions to reduce the calculation time duration are possible. The interface between the automation system, in this case the simulation environment, and IAV Kasai is an asynchronous interface. Hence, the test design algorithm is executed in a separate task, allowing the server to interrupt or communicate with the test point search. Once a test point is demanded but no solution is present, the test point search could be interrupted with taking the currently checked space-filling test point as next test point, even though it could be part of an exclusion area. A loss in test design quality would be the outcome but due to the adaptive design taking all measured and planned test points into account it

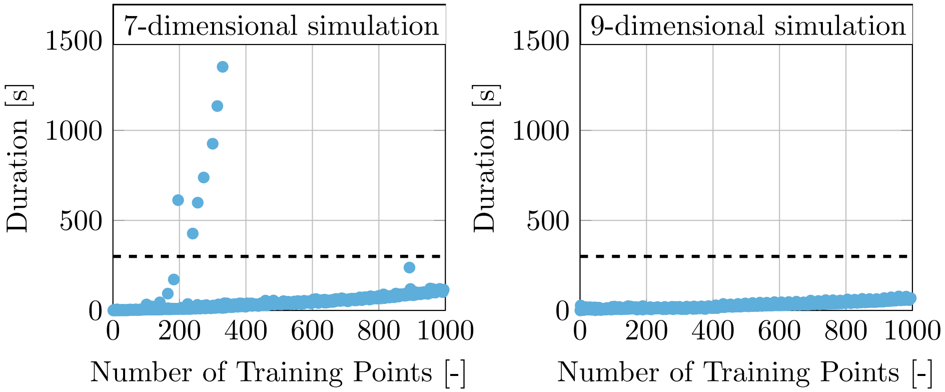


Figure 5.14: Calculation duration per test point during the 7-dimensional diesel engine simulation (left) and the 9-dimensional artificial model simulation (right) within the last simulation run for the adaptive multicriterial test design

still would be beneficial compared to a non-adaptive test design. Another reasonable solution is to switch the candidate erase strategy. Two different approaches are introduced in chapter 3 to identify candidates being part of the exclusion area. Within this simulation, the iterative exclusion procedure (section 3.3.2) is applied. This procedure has the main advantage that it can be utilized in high-dimensional input domains and with a high amount of convex hull-building test points. However, if the effort to calculate all convex hull hyperplanes is feasible within the given time duration, the hyperplane-based exclusion procedure (section 3.3.1) provides a faster solution. Especially for a low amount of hull training points the hyperplane-based exclusion procedure could be applied, switching to the iterative approach once the convex hull calculation duration exceeds a limit. This solution is only applicable to input dimensions smaller than nine, as the hull calculation duration will exceed the iterative solution from that point on (refer to figure 3.4 in section 3.2).

5.3 Chapter Summary

The four open research topics

- incorporate piecewise-defined limit behavior,
- find an optimal adaptive test design strategy,
- deal with more than one output and
- achieve a continuous measured input domain volume increase

are individually examined in the preceding chapter 3 and chapter 4. This chapter dealt with the development of a joined test design strategy that incorporates the findings regarding the treatment of engine boundaries and model quality improvement methods. The test design process starts with an initial phase to generate some necessary knowledge about the design domain. A small initial test plan is created by a volume maximization algorithm, considering configured input domain restrictions by a calibration engineer. Once the initial test phase is completed, a continuous test design strategy takes place. This design strategy combines both, the adaptive test space restriction strategy from chapter 3 and a model-based test design. As an outcome from chapter 4, the mutual information criterion was considered as the most powerful design method. However, the relevance-based test design strategy, which performs second-best, was applied to the combined test design strategy for calculation duration reasons. This test design process was integrated into a user-friendly environment on the base of the commercial software IAV Kasai that provides the necessary test design and modeling framework. The implementation additionally made possible a scheduling of calculation tasks to use the available idle time slots for all necessary calculations and prioritize the tasks to avoid unnecessary waiting times at the engine test bench. An adaptive validation point planning strategy as well as a method to involve user-specific sorting options was implemented as well.

The framework was tested in a simulation environment with two different simulation models. A 7- and a 9-dimensional global test were performed with the combined adaptive approach as well as with a non-adaptive space-filling test design planned in advance of the simulation. The comparison of the resulting model quality trend showed a significant improvement of all outputs in both simulations for the adaptive approach. A reduction of the area under the curve (AUC) value for the model error trend highlighted an improvement of the arithmetic mean over ten simulation runs between 15.5 and 60 percent. Additionally, the measured input domain volume significantly increased during the whole simulations for the adaptive approach. An increase in hull volume of 20 % for the 9-dimensional test and an increase of 110 % for the 7-dimensional simulation was the outcome of the simulation runs.

The adaptive validation point design and the non-adaptive design were compared regarding their ability to predict an estimated true average validation error within the particularly measured input domain. On average, the adaptively planned validation design significantly improved the validation RMSE validity compared to the validation points planned in advance. However, a deterioration was present for the soot output of the 7-dimensional test case, which originates from high output gradients near the boundary and hence a wrong test error estimation in these areas for the adaptively planned points. Additionally, the measured input domain volume was twice as large for the adaptive approach but with the same amount of validation points. The 9-dimensional test did not show such behavior, because the input domain was not as restricted as within the 7-dimensional test. Considering the higher measured input domain volume for the adaptive approach, the proposed validation point design is able to improve the error estimation. For strongly restricted test domains and outputs with high gradients near the boundary the error estimation generated by this test design has yet to be judged carefully.

From a calculation duration point of view, nearly all test point calculations stayed below the limit of 300 seconds duration. Within the 7-dimensional test, seven test point calculation processes did exceed the time limit due to the strong input domain restriction. This would lead to an interruption of the measurement process within such a test case but does not abort the process. However, a reduction of quality improvement per test time could be the outcome. Since the scheduled calculations in IAV Kasai could handle an interruption, possible solutions to avoid a waiting time with only little design quality deterioration were proposed.

6 Validation at a Combustion Engine Test Bench

Within this research, several new methods were developed and compared. All comparisons were based on a simulation model that focuses to reproduce the result of an engine test as best as possible. The model estimates the process behavior as well as the noise as it occurs at several measurement process stages. The simulation model has several advantages but a main disadvantage though. The main advantage of the simulation model is the possibility to test all introduced methods at very low cost compared to engine test bench measurements. All test runs could be executed several times to estimate the average result over several measurement campaigns and to judge the fluctuation between each campaign and for each method. The main disadvantage of the simulation model is the difference between the model outcome and a real engine test outcome. The most influencing parts of the model are the process model itself as well as the process noise, which are both difficult to estimate correctly.

For these reasons, the methodology research result is tested in a test bench environment. This test shall give rise about the overall quality improvement the developed methods provide. First, a test is designed regarding the system inputs to vary, the outputs to be modeled, the limits to be considered, and the conditions the engine is operated in (section 6.1). The subsequent section 6.2 contains the measurement results, model quality evaluation, and a quality comparison of an optimization problem solution for both methods. The findings are summarized in section 6.3.

6.1 Test Definition and Setup

For the comparison, a small-sized passenger car turbo charged four-stroke gasoline engine with high complexity in terms of input parameters is provided. The multicriterial adaptive design, as is introduced in the preceding chapter 5, is conducted for the test and is compared to a non-adaptive space-filling test design calculated with IAV Kasai. A local test without varying speed and load is executed. The test focus is to observe the transition area from naturally aspirated to turbocharged mode, meaning a relative in-cylinder air mass of 1, at a representative engine speed of 2000 rpm. Seven different parameters, shown in figure 6.1, are varied in a non-restricted hypercubic input domain. The input parameters are not named for reasons of confidentiality, but the author declares that this has no effect on the following results. The eight outputs to be modeled are

- coefficient of variation (COV) to judge combustion stability [Hey88],
- 50 % mass fraction burnt (MFB50) to identify the center of combustion [Hey88],
- exhaust temperature before turbine ($\mathrm{T}_{\mathrm{Bef\,Turbine}}$),
- engine torque (Torque),
- brake specific fuel consumption (BSFC) [Hey88],
- soot mass (MS),
- particulate number (PN) and
- waste gate position (WG).

Several engine limits take place that restrict the input domain. These are

- temperature after compressor,
- coefficient of variation,
- latest-allowed center of combustion,
- misfire and
- waste gate position

where the waste gate position is restricted to a minimum amount. Since the EGR rate is varied as one input, the waste gate has to be closed to ensure the same in-cylinder air mass especially in the observed transition area. If the EGR rate is raised while the waste gate is fully closed already the air mass would drop, which has to be prohibited. The spark timing is

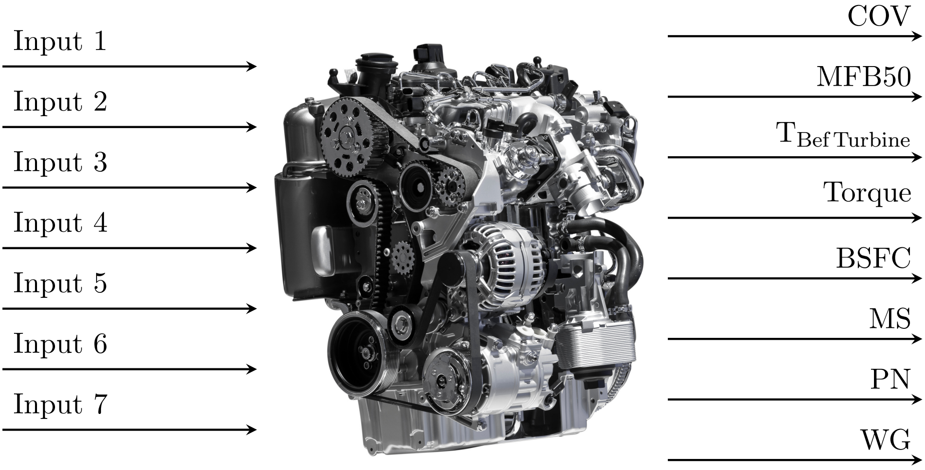


Figure 6.1: Combustion engine test definition with seven engine inputs to be varied and eight engine outputs to be modeled

automatically controlled during the whole test to a center of combustion of 8 °CA or, in case of knocking occurs, to the knocking border. 305 training points are planned for both methods. 36 validation points are measured by the non-adaptive approach and 10 % validations for the adaptive approach. Since the adaptive measurement is proceeded unmanned without a restriction for the maximum amount of measurements, a total training point count of 345 measurements and 34 validation points is recorded within the adaptive approach measurement. The first 305 training points are extracted and all other training measurements are used as additional validation points. In addition, the validation points of the non-adaptive measurement are added to the adaptive approach and vice versa. After erasing outliers and adjusting the amount of training points to be equal, both methods exhibit 297 training and 109 validation measurements. The validations are not used for model training but only to judge the average model prediction quality.

The engine in test is equipped to a combustion engine test bench. The ECU is controlled by the software ETAS INCA [ETA20b], the engine dyno control and the engine monitoring is performed by AVL PUMA Open™, which is the former version of AVL PUMA Open 2™ [AVL20]. The test automation is performed by A&D ORION, which connects to both AVL

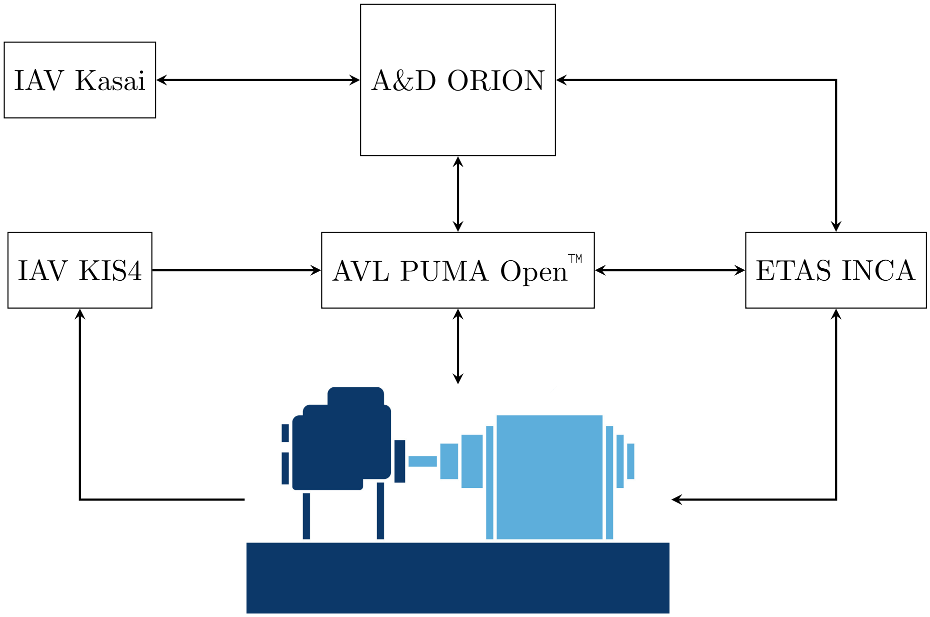


Figure 6.2: Schematic representation and interaction of the relevant systems at the engine test bench. The shown modules are only an extract of all necessary modules for the test, but they represent the main structure and communication channels.

PUMA Open™ as well as ETAS INCA to set the ECU parameters on the one hand and observe measurement channels to react to engine limits on the other hand. A connection between IAV Kasai and A&D ORION enables the automation system to retrieve a next test point information on demand and send the measurement result to the test design system. The developed interface setup as shown in figure 5.6 in chapter 5 is utilized for this test. The engine is equipped with in-cylinder pressure sensors at each cylinder. The data processing of the pressure sensors and knock evaluation of the processed data is done by the IAV KIS4 system. A test setting overview is given in figure 6.2. For the non-adaptive approach, the exact same setting is set-up without the necessity to connect IAV Kasai with A&D ORION. The pre-calculated space-filling test plan is applied to A&D ORION, which performs a measurement for each row sequentially. Both the EGR rate as well as the MFB50 are controlled by A&D ORION to the desired value, where all other parameters directly access actuators. Each test point is set vectorial with limit observation. In case a limiting boundary is reached, a surrogate measurement is executed near the boundary.

6.2 Results

The comparison of the non-adaptive and the adaptive multicriterial approach are compared in two different ways. First, the model quality trend as well as the convex hull volume growth are analyzed. Both criteria are significant to achieve models that are valid in a wide range with a high quality prediction ability. In a second step, an optimization problem is solved at different measurement phase steps, where the result is validated at the test bench to compare the outcome quality.

6.2.1 Model Quality and Measured Domain

In a post-processing step, all models are trained for each measurement step but with rating the model quality with all measured validation points. This is done for both procedures, which enables the comparison of both methods regarding their model quality evolution. Additionally, the convex hull of all training points is calculated to rate the measured input domain trend. For model training the IAV Kasai GPM modeling algorithm with automatic box cox transformation is used, which is a very common way for a calibration engineer to train models. The validation error is normed to the mean plus standard deviation of all measured data. A convex hull is calculated at each step by the data normed to $[0,1]^7$ with the complete measured data range as bounds.

The convex hull volume evolution for both measurement campaigns is shown in figure 6.3. Contrary to the expectation that the adaptive approach volume at least develops at the same level as the non-adaptive approach, during the first 100 measurements a lower volume is present for the adaptive approach. This is mainly due to the reason that both the test points as they are planned and measured boundary points are considered during the test design. The non-adaptive approach only considers test points and is not able to incorporate measurements to the test design. This procedure on the one hand increases the space-filling quality but on the other hand could lead to a lower hull volume, as is present in this measurement. However, starting from approximately 60 measurements the adaptive procedure

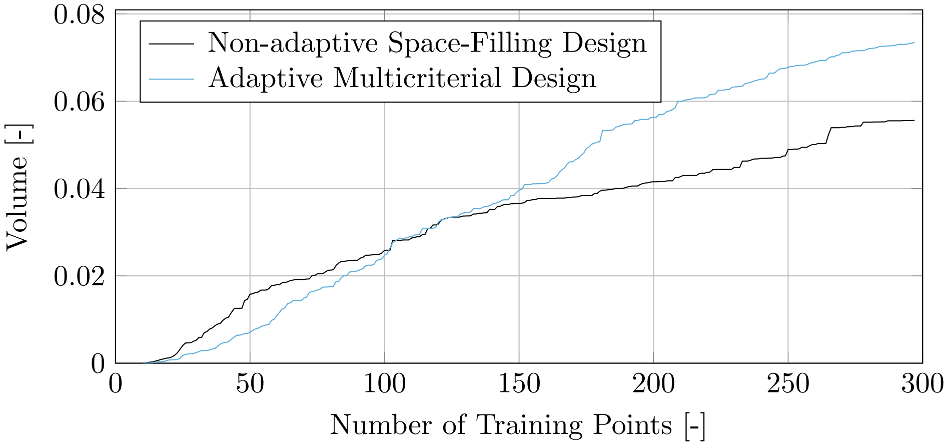


Figure 6.3: Convex hull volume trend during the measurement campaign for both test design methods. A higher volume states a larger measured area and is to be preferred. The maximum volume of 1 is the highest possible value if no limits are present and the convex hull fully includes the test space.

demonstrates its advantage and reduces the candidate set by the prediction of immeasurable areas. The convex hull volume increase rate rises and is on a high level nearly during the entire measurement campaign, whereas the non-adaptive hull volume increase rate declines. At the end of the measurement, a nearly 34 % higher volume is present for the adaptive approach, which is a significant increase at a similar measurement duration level.

The model quality difference, rated by the validation point set, is shown in figure 6.4. An AUC value is calculated for each trend and for both methods, providing a quality rating for the whole measurement campaign. The AUC value for each particular output of the adaptive result is set in relation to the non-adaptive result and the difference is calculated to obtain a deterioration or improvement value for the adaptive approach. A significant increase in model quality is offered for all outputs with an exception for the particle emission model. From an overall point of view, an average model quality improvement, rated by the AUC criterion, of 11 % is achieved despite the deterioration of the model quality of one output. Considering the relative model error at the end of the measurement campaign, a mean model error decrease by 17 % is achieved over all outputs. In addition to the measured volume increase, this offers a good possibility to identify

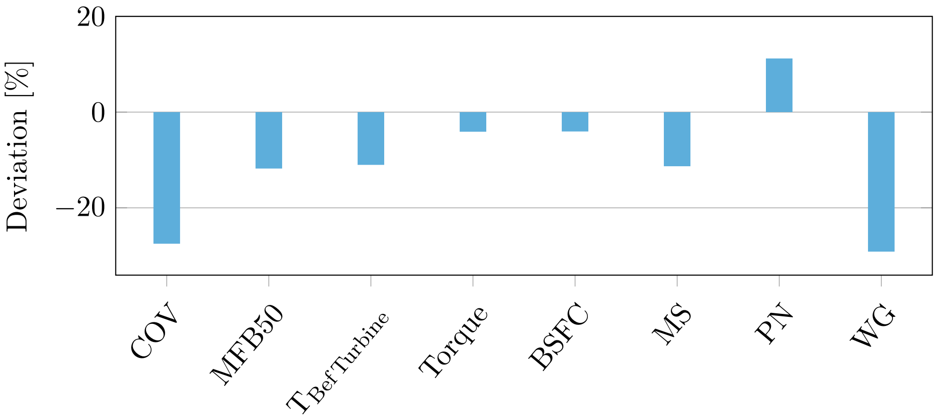


Figure 6.4: Relative AUC criterion for each output, calculated for the validation RMSE over the entire measurement campaign. The relative deviation of the AUC value of the adaptive multicriterial approach to the non-adaptive space-filling design is shown. A negative value is an improvement in model quality, whereas a positive value shows a deterioration by using the adaptive approach.

the engine behavior more accurately in a larger range with exactly the same amount of test bench measurement time. The deterioration in model quality for PN, however, is an undesirable outcome but could have several reasons. The PN emission model could suffer from the relevance-based test design, which could be supposed due to the worsened soot model within the simulation investigation study, see figure 4.13 for reference. The soot mass model (MS) is the reason why this is not likely. The soot mass and particle emissions in a gasoline engine are strongly correlated. While the soot mass model offers an 11 % improvement, the particle model shows an 11 % deterioration. A more likely reason to this is the poor reproducibility of particle measurements. A hint to this relation is given by figure 6.5, where the normalized validation error trend is shown for the particle emissions (left) and for comparison for the MFB50 output (right). While the MFB50 model shows a smooth decrease in model error at a significantly lower relative error with clearly better performance of the adaptive approach, the particle model error develops roughly from measurement to measurement. This shows the high uncertainty in model training, due to a high process and measurement noise. However, both approaches have a similar model error evolution starting from 140 training points and reach nearly the same level with the full training data. This is different for all other outputs, which show a lower model error for the adaptive approach at the end of the measurement campaign. All remaining model error trends for both methods can be found in appendix F.

The improvement of the model quality and of the measured input domain volume increase by using the adaptive multicriterial test design were similar to the simulation-based results, shown in chapter 5. Figure 6.6 shows a box plot for all AUC values (lower box) and model test errors (upper box) gathered by the adaptive design in the 7-dimensional simulation as described in section 5.2 with 295 training points. The values are given as the normed deviation to the non-adaptive design results. The dots do not show outliers but represent the deviation of the normed values for the models trained by the test bench measurements. The test design influence on the test bench results mostly corresponds to the simulation outcome, which indicates the validity of the simulation model introduced and applied in this thesis. An exception is present for the AUC value of the PN, Torque and BSFC model exceeding the simulation result range. The deviation between the simulation-based and the measurement-based outcome can be explained by two reasons. Firstly, the validation point sets offer different properties, which has a high impact on the test error validity. The number of validation test points and the point set distribution are very different between simulation and measurement. Additionally, the validation points used during simulation were gathered noise free, while the measured validation points at the test bench observe the same noise as the training points. A second reason is the utilized noise model that does not discriminate between the outputs. The applied relative noise level was assumed to be equal for each output. Both of these influences are responsible for the deviation but the results still offer a comparability regarding the test design method influence.

6.2.2 Optimization Problem

A very practical investigation to judge the outcome of the adaptive multicriterial approach is to solve an optimization problem. A main target in engine calibration is to find a setting with lowest possible fuel consumption to meet the legislation targets regarding CO_2 emissions. At the same time, other exhaust emissions may be taken into consideration, but a limit for

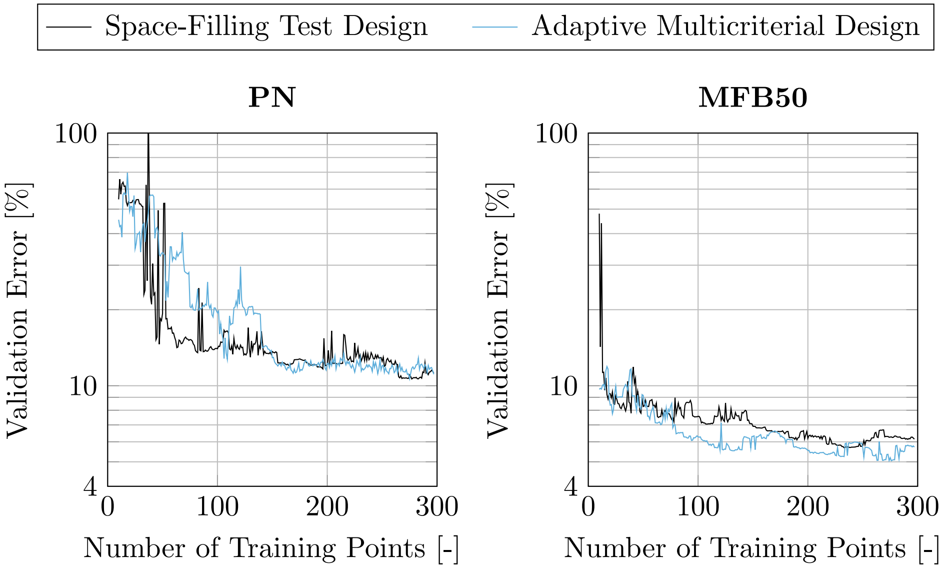


Figure 6.5: Model quality trend given the amount of training points, assessed by the normed validation RMSE for the outputs particulate number (left) and the center of combustion (right) for both, the non-adaptive and the adaptive approach.

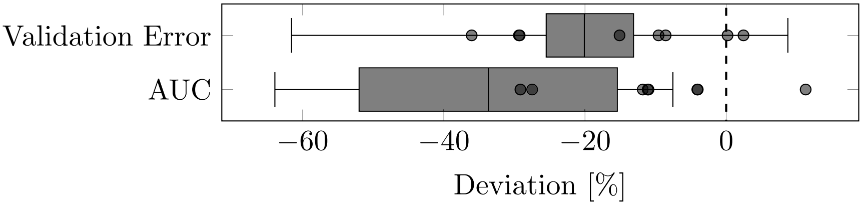


Figure 6.6: Box plot representing the model validation error AUC value (lower box) and the model validation error (upper box) each with 295 training points for the 7-dimensional simulation result introduced in chapter 5. The dots represent the results of the test bench measurement and indicate the comparability between simulation and test bench measurement. The data is calculated as the relative deviation of the model quality given the adaptive and the non-adaptive test design results for each output and for all simulation repetitions. A negative value corresponds to an improvement by the adaptive test design method.

the given operating point is not present in this case. Additionally, the emissions are commonly considered during a global map optimization, e.g. by meeting accumulated limits. The optimization problem to be solved here is an example to find a suitable setting with lowest possible fuel consumption. However, as a constraint the waste gate position has to be considered as it must not be lower than a defined threshold given the input setting. In case it would be fully closed, an engine load increase would not be possible by closing the waste gate anymore and therefore this constraint has to be met. This problem can be solved by a gradient-based optimization procedure considering the relevant modeled outputs. The global map optimization tool IAV OptiMap, which is part of IAV Kasai, is used to find an optimal setting for lowest fuel consumption fulfilling the waste gate constraint. The optimization problem can be formulated as

$$\min_{\boldsymbol{x} \in \mathrm{co}(\boldsymbol{X})} \mathrm{BSFC}(\boldsymbol{x}) \text{ such that } \mathrm{WG}(\boldsymbol{x}) \geq t \tag{6.1}$$

given the threshold t. Another study is conducted to rate the model quality at an earlier state of measurement. Therefore a model is trained with 200 training points for both methods and the same optimization problem is solved. All four different optimized settings are measured at the test bench and compared.

A slightly worse result of the BSFC is present for the optimization by the adaptive approach (figure 6.7 left). A deterioration of 0.6 % and 2 % in relation to the non-adaptive approach is the result for the model with 200 and 297 training points, respectively. It could be assumed that the deterioration of BSFC over training points for the adaptive approach is a result of a worse model prediction compared to the non-adaptive one. However, considering the result of the constrained WG position (figure 6.7 right) explains the higher deviation. While the waste gate position criterion is hardly met for the non-adaptive approach in both measurements, a reduction from 10 % to 5 % deviation from the target is given for the adaptive approach with increasing number of training points. Neither a good result, nor a clear improvement over training points is the outcome for the non-adaptive approach.

This optimization problem is a showcase that demonstrates the necessity of a good overall model quality for several outputs. In engine calibration, more complex optimization problems have to be faced, especially for map

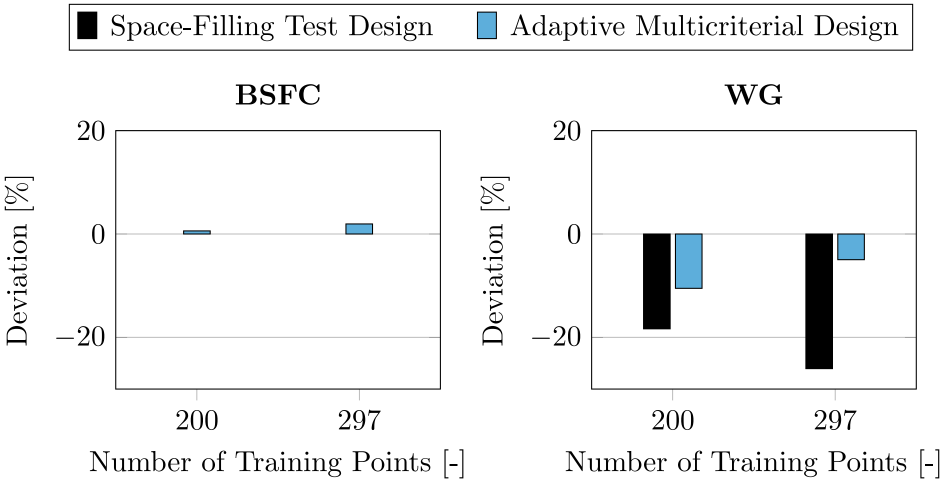


Figure 6.7: Measured optimization results for both, the adaptive and the non-adaptive approach. Left: Result for the fuel consumption. Since the reference is the non-adaptive space-filling approach, only the adaptive multicriterial design is shown. Right: Measured deviation of the waste gate position in relation to the optimization constraint t.

optimizations in the whole engine operation area and with several more outputs to be considered.

6.3 Chapter Summary

In this chapter, a real-world-based validation of the methods developed is presented. The multicriterial adaptive test design approach, which is based on a combination of a model focusing and an unknown region test design, was applied to an engine calibration problem with 7 different input parameters to vary. The same problem was measured with a non-adaptive space-filling test design. The implementation of the multicriterial design to IAV Kasai and the developed interface to A&D ORION was applied at an engine test bench to execute the adaptive test design procedure. The validation measurements of both approaches were combined to assess the model quality trend by 109 measurements. Both the overall model quality for 8 different outputs and its measured input domain was compared as well as an optimization problem was solved and reviewed by measurements.

An increase of the measured input domain by 34 % was present with all measurements by using the adaptive test design approach. The average model quality increase trend, rated by the AUC criterion, showed a mean improvement of 11 % over all eight outputs. When observing just the outcome with all measured training points, an increase of model quality by 17 % was the outcome.

The optimization of the fuel consumption, given the waste gate position as a constraint, showed the importance to achieve a high model quality for several outputs. A significant difference regarding the observance of the waste gate criterion could be noted, with an up to 21 % better matching for the adaptive approach at an only 2 % deterioration of fuel consumption.

All results except the model quality trend of the particulate number output could be optimized compared to a non-adaptive space-filling approach. The AUC value for the PN model, rating the model quality improvement progress, showed an 11 % deterioration for the adaptive approach but with the same resulting error level by training the model with all data. Since this study conducted all measurements only once due to the high test bench costs, a result validation and especially the determination of the method influence on soot emissions has to be conducted in further studies. Especially soot emissions tend to show very high process and measurement noise, which is why a single result is not robust regarding the effect of a test design. However, a comparison to the simulation-based results offered a good comparability, showing a similar relative model quality improvement.

7 Conclusion and Outlook

This thesis deals with the adaptive test design methodology within the field of steady-state combustion engine calibration. The main focus was the development of a user-friendly adaptive test design process that is able to handle restrictions in the design domain and at the same time reduces the test bench measurement effort compared to non-adaptive procedures. The criteria to judge the outcome of a measurement campaign were defined to be the covered input domain to prohibit model extrapolation as well as the average model quality. A convex hull volume gave rise about the covered area of the measured test design. Model quality was judged by the RMSE of an additional test measurement set in the restricted input domain, comparing model prediction and the true process outcome within a simulation environment and additional test measurement data within a test bench measurement campaign. An additional weak criterion for the method development was the test duration. An adaptive test design procedure should not extend the measurement time per test point. With these design criteria, the following topics were addressed.

Measured domain volume increase

Many existing adaptive test design strategies do have in common that they utilize either a regression model or a tunable geometrical model. A main disadvantage of a regression model is the missing ability to represent piecewise-defined measurement channels like knocking or misfire. The convex hull approach as a non-tunable deterministic model, which is often used during optimization to prohibit a trained model from extrapolation, is not applicable within an adaptive test design strategy, because it does not permit an enlargement of the measured region. Therefore, the two research topics

- how to incorporate piecewise-defined limit behavior and
- achieve a continuous measured input domain volume increase

were addressed. To provide a solution to these topics, an extension of the convex hull model was developed. Measured limit points are treated as hard limits in the input domain, whereas hull boundary points not labeled as limit points are extended geometrically without any tunable parameters. This approach includes all theoretically possible convex areas within the design domain into the hull. Two different solutions were introduced that present a fast calculation in low dimensional problems by calculating the exact convex hull given its hyperplanes and use linear algebra methods to extend the design domain. In case the number of input parameters exceeds six, an iterative test point search is applied without calculating the exact design domain. The convex hull test is performed by solving the convex set solution by means of the GJK algorithm [GJK88], whereas the hull extension is derived by a modified convex cone algorithm. A low-discrepancy test design was applied in the restricted but extended area that gives a compromise between model quality enhancement for a GPM and at the same time increases the measured input domain volume due to its characteristic to plan test points far away from existing ones.

The newly developed strategy was applied in a simulation environment comparing the new strategy with a non-adaptive low-discrepancy test design. The new strategy provided a 45 % larger measured input domain volume in a 9-dimensional test case and a doubled volume in a 7-dimensional test case. The model quality slightly improved as well, which in combination offers a huge benefit for a subsequent ECU setting optimization.

Model quality optimization

Since engine test bench measurements are very cost intensive, it is an ongoing requirement to reduce the necessary amount of measurements during the engine development process as far as possible. With the focus on a GPM, the objective was to find a most beneficial test design strategy for the optimization of the model quality with respect to the amount of measurements. Another common requirement within model-based calibration is to consider several outputs. Applying an adequate test design strategy therefore has to fulfill the requirement to optimize several models at the same time. To the state of the art, typical procedures were to optimize several outputs in a batch mode or with a round-robin procedure. From these requirements, the two research topics

- to find the optimal adaptive test design strategy and
- how to deal with more than one output

were derived.

To find the optimal test design strategy, the most promising strategies from literature, in particular the maximum entropy and the mutual information test design, were considered and compared to a newly developed mixture of low discrepancy and maximum variance as well as a relevance-based test design. Since the mutual information test design is in its defined setting not applicable to problems with a high amount of possible test point candidates, a simplification was introduced and used for the comparison. Based on the different test design strategies, a combined test design for several outputs was developed for each algorithm, providing a non-parametric, simultaneous optimization for all outputs. These design strategies comply with the objective to find a test point that maximally contributes to all outputs. As an exception, the solution for the variance-based design generates a test point for the worst model only, which is equivalent to a round-robin procedure.

As a result from a simulation including all strategies with a combined output optimization, the simplified mutual information test design outperformed all other designs. The relevance-based test design however provided the second-best results, but for a significantly lower computational complexity. The variance-based and the entropy-based test design showed disappointing results with even worse performance than a non-adaptive low-discrepancy design in some test cases. Compared to a non-adaptive low-discrepancy test design, the model quality improvement for the modified mutual information design was found to be 21 % on average over all outputs of a seven- and a nine-dimensional test case, whereas the relevance-based test design still shows an improvement of 15 %.

Multicriterial approach

To obtain a user-friendly test design and measurement process including both measurement domain volume increase as well as model quality optimization, a multicriterial test design strategy was developed. Based on a low-discrepancy test design method, with each test point a decision is made if it should mainly contribute to a volume increase or model quality improvement. With this procedure, an automatic decision can be made which test design method should be applied next to obtain a continuous volume increase and model error reduction without any user intervention. The extended convex hull boundary model as well as the relevance-based test design are incorporated into this strategy. The relevance-based test design was chosen for calculation performance reason. However, the model quality focused design could be easily exchanged within the multicriterial strategy due to its modular architecture. To comply with all requirements of a model-based calibration process, further topics were addressed. Since the test plan is not completely calculated in advance of the measurements, a validation point design strategy is applicable. Additionally, a sorting of test points to accelerate the engine tests was necessary and therefore was implemented. The test design strategy was integrated into the commercial model-based calibration software IAV Kasai with an open interface to be applicable to any type of automation system. With this integration, a compromise to satisfy all four open research topics simultaneously was achieved.

The multicriterial design strategy was applied in a simulation environment, simulating a global measurement including engine speed and load for testing its ability to cope with a common meander sort option. The results were compared to a non-adaptive test design with a sorted test plan. An overall model quality improvement of 36 % was the outcome over two different simulation configurations with four outputs each. The convex hull volume increased 110 % for a seven-dimensional test and 20 % for a nine-dimensional test case in addition to the model quality improvement. In most cases, the calculation duration per test point was significantly lower than the target of 300 seconds. However, in case the input domain is strongly restricted, the test point search duration increases, which led to 0.7 % of the test points within the strongly restricted seven-dimensional test case to exceed the limit. The comparison of the validation point settings showed a diverse result. While the validation error within the relevant area was mostly clearly improved by the adaptively planned validation point design, especially outputs with strong gradients near the boundary were found to be sensitive regarding the error estimation. The newly developed design puts a higher weight on validation points inside the measured domain, which is why the boundary-near estimation suffers from this design. Excluding this effect, a significant improvement of the true validation error estimation was achieved planning the validation points adaptively.

In a real world example at the engine test bench, varying seven different inputs of a gasoline engine without speed and load showed similar results as the simulation. The model quality evolution of eight out of nine outputs was optimized up to 27 % compared to a non-adaptively planned measurement campaign. A slight deterioration was present for the particle emission model quality trend but with a similar resulting model quality using all measurements. The hull volume exhibited a 34 % larger measured region applying the multicriterial design strategy. The resulting models were used to solve an optimization problem, where the better results of the adaptively planned strategy were confirmed.

Outlook

On the base of the introduced adaptive test design strategy, some improvements could be investigated in future work:

- **Focused hull enlargement:** The combination of a geometric exclusion area as introduced with the prediction of possible boundaries in the model extrapolation area would enable the test design algorithm to selectively plan test points outside the current measured domain but with the focus on hull enlargement only. This strategy could combine the safety of the convex region approach incorporating piecewise-defined limit behavior with the flexibility and continuous prediction of the regression model approach shown in [Sch+18]. However, the model quality focus could disappear. Thus the model quality and the hull volume increase have to be judged in common carefully.
- **Biased model quality improvement:** In advance of a measurement campaign, the calibration engineer mostly knows which outputs are hard to model and which are most important to solve the optimization problem. A biased test design could be investigated that aims at optimizing some outputs more than others, either by the engineer's choice or by the model quality. The relevance-based test design as introduced in this thesis puts the same weight on the length scales of each output. However, a non-uniform weighting could be applied to improve some models more than others without introducing a round-robin or batch mode optimization. A similar approach could be applied to the mutual information test design, which is investigated with a uniform weighting over all outputs as well. A bias could

be applied to change the information content of each candidate by user's choice or by a model quality rating.

- **Calculation performance improvement:** With the proposed design, an interruption in the measurement process could appear, especially in strongly constrained input domains. Some simple procedures are proposed to provide a test point with reduced quality on demand. However, different solutions could be investigated as a compromise between quality reduction and time loss. Since the convex cone strategy offers the part with most computational complexity, it could be exchanged by a model prediction strategy in case the automation system demands a next test point. For test cases without piecewise-defined limit channels, a general combination of the mutual information test design with a regression model-based boundary model is another approach to reduce calculation duration and incorporate the mutual information test design.

A GJK Algorithm

The base equation for the GJK algorithm as introduced in [GJK88] is the definition of the convex hull co() of a given point set $\boldsymbol{X}$ with size k in a domain of dimension dim by

$$\operatorname{co}(\boldsymbol{X}) = \left\{ \sum_{i=1}^{k} \alpha_i \boldsymbol{x}_i \,\middle|\, \sum_{i=1}^{k} \alpha_i = 1 \text{ and } 0 \leq \alpha_i \leq 1 \;\forall i \right\}. \tag{A.1}$$

That is, a given test point $\boldsymbol{t}$ is checked if it is part of the convex hull if any combination $\boldsymbol{\alpha}$ is found, which fulfills the constraints and represents $\boldsymbol{t}$ if (A.1) is applied.

The GJK algorithm is used to find a non-unique solution for $\boldsymbol{\alpha}$. It assumes the test point to be the origin, which is why the point set $\boldsymbol{X}$ is shifted by the test point $\boldsymbol{t}$ to $\boldsymbol{X}_t$. A point on the surface of the convex hull is determined iteratively, which shows least distance to the origin. The algorithm consists of two main calculations steps. A simplex is created by the selection of maximum dim + 1 points from $\boldsymbol{X}_t$. For the selection, a support value is assigned to each hull vertex by a support vector $\boldsymbol{v}$. The starting support vector can be random. In this implementation, it is used as the vector starting at the test point and pointing to the center of gravity of the hull building points

$$\boldsymbol{v} = \operatorname{mean}(\boldsymbol{X}) \tag{A.2}$$

which is derived only by the center of gravity, since the test point is the origin. The support values are the distances from the test point to the perpendicular projection of the hull vertices on the support vector by

$$s_i = \boldsymbol{x}_t^\top \boldsymbol{v} \;\forall\; \boldsymbol{x}_t \in \boldsymbol{X}_t. \tag{A.3}$$

The vertex with lowest support $\min(\boldsymbol{s})$ is added to the simplex generating vertices $\boldsymbol{V}_j$ in iteration j. The second step is the distance calculation of the origin to the current simplex $\boldsymbol{V}_j$. To calculate the minimum distance, all 1...(dim+1)-dimensional elements of a simplex have to be considered, which means all vertices, lines, planes, etc. In [GJK88] the so-called Johnson distance algorithm [Joh87] is proposed. A different way is to define a set of linear equations for each element, as derived in [Cam97], which can be solved. The linear equations are created by the assumption that the closest point has to be perpendicular to all lines within the considered object. That is, if the object is a 2-dimensional simplex described by 3 points $\boldsymbol{x}_1, ..., \boldsymbol{x}_3$, the perpendicular projection of the origin on the plane is defined by two linear equations

$$\begin{aligned} (\boldsymbol{x}_2-\boldsymbol{x}_1)\boldsymbol{x}_1\alpha_1 + (\boldsymbol{x}_2-\boldsymbol{x}_1)\boldsymbol{x}_2\alpha_2 + (\boldsymbol{x}_2-\boldsymbol{x}_1)\boldsymbol{x}_3\alpha_3 = 0 \\ (\boldsymbol{x}_3-\boldsymbol{x}_1)\boldsymbol{x}_1\alpha_1 + (\boldsymbol{x}_3-\boldsymbol{x}_1)\boldsymbol{x}_2\alpha_2 + (\boldsymbol{x}_3-\boldsymbol{x}_1)\boldsymbol{x}_3\alpha_3 = 0 \end{aligned} \tag{A.4}$$

with the restriction of only positive values for all α_i. Another restriction is $\sum_{i=1}^{3} \alpha_i = 1$ which forces the perpendicular projection being within the bounds of the simplex and not anywhere on the plane. With these 3 equations in matrix notation the linear equation set

$$\begin{bmatrix} 1 & 1 & 1 \\ (\boldsymbol{x}_2-\boldsymbol{x}_1)\boldsymbol{x}_1 & (\boldsymbol{x}_2-\boldsymbol{x}_1)\boldsymbol{x}_2 & (\boldsymbol{x}_2-\boldsymbol{x}_1)\boldsymbol{x}_3 \\ (\boldsymbol{x}_3-\boldsymbol{x}_1)\boldsymbol{x}_1 & (\boldsymbol{x}_3-\boldsymbol{x}_1)\boldsymbol{x}_2 & (\boldsymbol{x}_3-\boldsymbol{x}_1)\boldsymbol{x}_3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{A.5}$$

is defined and can be solved by left division. The general equation for a dim-dimensional simplex with (dim+1) vertices is

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ (\boldsymbol{x}_2-\boldsymbol{x}_1)\boldsymbol{x}_1 & (\boldsymbol{x}_2-\boldsymbol{x}_1)\boldsymbol{x}_2 & \cdots & (\boldsymbol{x}_2-\boldsymbol{x}_1)\boldsymbol{x}_{\mathrm{dim}+1} \\ \vdots & \vdots & \ddots & \vdots \\ (\boldsymbol{x}_{n+1}-\boldsymbol{x}_1)\boldsymbol{x}_1 & (\boldsymbol{x}_n-\boldsymbol{x}_1)\boldsymbol{x}_2 & \cdots & (\boldsymbol{x}_n-\boldsymbol{x}_1)\boldsymbol{x}_{\mathrm{dim}+1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{\mathrm{dim}+1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{A.6}$$

which solution is valid only if $\alpha_i \geq 0 \; \forall i = 1...(\mathrm{dim}+1)$.

In each iteration of the GJK algorithm, all elements of $\boldsymbol{V}_j$ are checked. The vertices of the element that contains the closest point are used for the subsequent iteration and added to $\boldsymbol{V}_{j+1}$. The algorithm stops if either the origin is part of the convex hull and therefore the distance from the closest point to the origin is zero, or if $\boldsymbol{V}_j = \boldsymbol{V}_{j-1}$. The numerical tolerance for the abortion strongly depends on the computational accuracy and the codomain of all given points. To find a universal tolerance, the points can be normed to a defined codomain, as e.g. $[0{,}1]^{\mathrm{dim}}$. Within this codomain, a robust numerical tolerance is found to be $\epsilon = 2.22 * 10^{-11}$. The full algorithm is shown in algorithm A.1.

Algorithm A.1: Algorithm to find the closest point on a convex hull for given hull points and a test point by a modified GJK algorithm

```
Input: Hull Points $\boldsymbol{X}$, Test Point $\boldsymbol{t}$, Numerical Tolerance $\epsilon$
Output: Closest Point $c$, Distance $d$
 1: Shift hull points $\boldsymbol{X}_t = \boldsymbol{X} - \boldsymbol{t}$
 2: Initialize $j = 1$, $\boldsymbol{V}_1 = \boldsymbol{V}_0 = \emptyset$
 3: Calculate support vector
 4: Calculate support values $\boldsymbol{s}$
 5: Add vertex with lowest support $\boldsymbol{V}_j = (\boldsymbol{V}_{j-1} \cup \mathrm{argmin}_{\boldsymbol{x}_t \in \boldsymbol{X}_t}(\boldsymbol{s}))$
 6: if $\boldsymbol{V}_{j-1} \equiv \boldsymbol{V}_j$ then
 7:     Set $\boldsymbol{c} = \boldsymbol{c}_{j-1}$, $d = d_{j-1}$
 8:     return
 9: end if
10: Calculate closest point $\boldsymbol{c}_j$ on $\boldsymbol{V}_j$ to origin by distance algorithm
11: Calculate Euclidean distance $d_j = |\boldsymbol{c}_j|$
12: if $d_j < \epsilon$ then
13:     Set $\boldsymbol{c} = \boldsymbol{c}_j$, $d = d_j$
14:     return
15: end if
16: Remove unnecessary vertices from $\boldsymbol{V}_j$ and set $\boldsymbol{V}_{j+1} = \boldsymbol{V}_j$
17: Increment j and return to row 2
```

B Convex Cone Algorithm

Given a point set $\boldsymbol{X}$, a convex cone area cone($\boldsymbol{X}$) including the origin is defined by a linear equation that is constrained by the parameter vector $\boldsymbol{\alpha}$

$$\text{cone}(\boldsymbol{X}) = \left\{ \sum_{i=1}^{k} \alpha_i \boldsymbol{x}_i \,\middle|\, \alpha_i \geq 0 \ \forall i \right\}. \tag{B.1}$$

A solution to $\boldsymbol{\alpha}$ must be found where each α_i must be greater or equal 0 if a test point $\boldsymbol{t}$ shall be checked for being part of the convex cone. Since this calculation is analytically not solvable, [ZC09] introduced an iterative convex cone calculation procedure. The algorithm is based on the GJK algorithm described in chapter A.1 and introduced in [GJK88], where a support vector is the basic element within each iteration. However, the support vector $\boldsymbol{h}$ differs in the convex cone algorithm from the GJK algorithm, because the origin is always part of the cone and the point set shift by the test point $\boldsymbol{t}$ does not apply. Therefore, the support vector is defined by the newly found closest point $\boldsymbol{c}_j$ in each iteration and the test point

$$\boldsymbol{h} = \boldsymbol{t} - \boldsymbol{c}_j. \tag{B.2}$$

A support value is derived for each cone vertex in the same way as in the GJK algorithm, while the point with the maximum support is added to the subset $\boldsymbol{V}_j$ respectively. The objective of the iterative algorithm is to minimize the maximum support value, where a solution is present if the maximum support converges to zero. While the GJK algorithm maximizes the support value, the convex cone algorithm is a minimization procedure, because the support vector originates at the newly found closest point $\boldsymbol{c}_j$ instead of originating at the point under investigation $\boldsymbol{t}$. The support vector is initialized to the point $\boldsymbol{t}$, whereas the subset $\boldsymbol{V}_j$ initially is an empty set. The algorithm can be summarized to the following steps:

1. Calculate support values $\boldsymbol{s}$

2. If the maximum absolute support value is less ϵ or the maximum support value did not change, $\boldsymbol{c}_{j-1}$ is the closest point and the algorithm stops
3. Add vertex with highest support to $\boldsymbol{V}_j$
4. Calculate closest point $\boldsymbol{c}_j$ on the open cone $\boldsymbol{V}_j$ to $\boldsymbol{t}$
5. Identify necessary vertices $\boldsymbol{V}_{j+1}$ as subset of $\boldsymbol{V}_j$
6. Calculate new support vector $\boldsymbol{h}_{j+1}$, increment j and return to step 1

Both algorithms, the GJK and the convex cone algorithm, are very similar but mainly differ in identifying the closest point $\boldsymbol{c}_j$ and the necessary vertices for the subsequent iteration in step 4 and step 5. For this calculation, a subalgorithm is introduced in [ZC09]. For the application in this thesis, the termination tolerance is set to $\epsilon = 2.22 * 10^{-9}$, which is more imprecise than the GJK tolerance due to a higher amount of calculations steps in the subalgorithm. The entire convex cone algorithm is described in algorithm B.1 in detail.

Algorithm B.1: Convex Cone Algorithm

```
Input: point set $\boldsymbol{X}$, considered point $\boldsymbol{t}$
Output: closest point $\boldsymbol{t}_c$, distance $dist$, isInside $bIn$
$\epsilon = 2.22 * 10^{-9}$
$j = 1, \boldsymbol{V}_1 = \emptyset$
calculate support vector $\boldsymbol{h}_1 = \boldsymbol{t}$
$\boldsymbol{s}_0 = \text{inf}$
while true do
    for all $\boldsymbol{x}_i \in \boldsymbol{X}$ do
        calculate support values $s_{i,j} = \boldsymbol{x}_i^\top \boldsymbol{h}_j$
    end for
    if $\max(\boldsymbol{s}_j) < \epsilon$ or $\max(\boldsymbol{s}_j) == \max(\boldsymbol{s}_{j-1})$ then
        $\boldsymbol{t}_c = \boldsymbol{c}_{j-1}$
        $dist =$ Euclidean distance $|\boldsymbol{t}_c - \boldsymbol{t}|$
        $bIn = \text{all}(\text{abs}(\boldsymbol{s}_j)) < \epsilon$
        return
    end if
    add hull point with highest support $\boldsymbol{V}_j = \boldsymbol{V}_j \cup \arg\max_{\boldsymbol{X}}(\boldsymbol{s})$
    Find closest point $\boldsymbol{c}_j$ on open cone $\boldsymbol{V}_j$ by subalgorithm given in
    [ZC09]: $[\boldsymbol{c}_j, relatedVertices] = coneDist(\boldsymbol{V}_j)$
    $\boldsymbol{V}_{j+1} = relatedVertices$
    $\boldsymbol{h}_{j+1} = \boldsymbol{t} - \boldsymbol{c}_j$
    $j = j + 1$
end while
```

C Exclusion Area Check

The exclusion area check is given by two different algorithms. In case the convex hull-building hyperplanes are present, a simpler calculation can be applied, because the joint hyperplanes of a hull vertex can be identified and the check for the exclusion area is based on the point hyperplane distance calculation. The full algorithm for the exclusion area check for given hyperplanes is shown in algorithm C.1.

Algorithm C.1: Exclusion area check for a candidate set given hyperplanes

```
Input: Candidates $\boldsymbol{S}$, Normal Vectors $\boldsymbol{A}$, Origin Distances $\boldsymbol{b}$,
       Hull Points $\boldsymbol{X}$, Boundary Labels $\boldsymbol{o}$
Output: $\boldsymbol{p}$
$\boldsymbol{p}$ = false[length($\boldsymbol{S}$)]
for all $\boldsymbol{x}_i \in \boldsymbol{X}$ do
    if not $\boldsymbol{o}[i]$ then
        continue
    end if
    find joint hyperplanes $\boldsymbol{A}_j, \boldsymbol{b}_j$ for $\boldsymbol{x}_i$
    for all $\boldsymbol{s}_k \in \boldsymbol{S}$ do
        calculate distances $\boldsymbol{d} = \boldsymbol{A}_j \boldsymbol{s}_k^\top - \boldsymbol{b}_j$
        if all $\boldsymbol{d} > 0$ then
            $\boldsymbol{p}[k]$ = true
        end if
        if all in $\boldsymbol{p}$ then
            return $\boldsymbol{p}$
        end if
    end for
end for
return $\boldsymbol{p}$
```

A more complex case is present if only the convex hull building point set with boundary labels is given. The hull hyperplanes are unknown and a calculation of those is not the objective for the exclusion area check. Therefore the convex cone algorithm as described in appendix B takes place. However, for each boundary point some precalculations are necessary to achieve the conditions to apply the convex cone algorithm. To test a point labeled as a boundary point of the given hull points, it has to be the origin, because the convex cone algorithm always applies the origin as cone starting point. Therefore all considered points are shifted by the boundary point. The cone then is spanned by all convex hull points, whereas the outside pointing cone is the desired one. This is achieved by mirroring the considered candidate $\boldsymbol{s}$ at the new origin. Applying the convex cone algorithm to find the closest point on the cone starting at the mirrored candidate offers the necessary information if the given candidate is part of the particular exclusion area. The full algorithm is detailed in algorithm C.2 and the preconditioning is exemplarily shown in figure C.1.

Slightly non-convex hull points lead to a misinterpretation if the convex cone algorithm is utilized. In this case, a repositioning has to be applied that projects the test point on the nearest hull-building hyperplane. The procedure for this repositioning is explained in algorithm C.3. Points that are labeled as boundary points but do not contribute to the convex hull can simply be identified by the GJK algorithm.

Algorithm C.2: Exclusion area check for a given candidate set by the convex cone algorithm

```
Input: Candidates $\boldsymbol{S}$, Hull Points $\boldsymbol{X}$, Boundary Labels $\boldsymbol{o}$
Output: $\boldsymbol{p}$
$\boldsymbol{p} = \text{false}[\text{length}(\boldsymbol{S})]$
for all $\boldsymbol{x}_i \in \boldsymbol{X}$ do
    if not $\boldsymbol{o}[i]$ then
        continue
    end if
    $\boldsymbol{X}_t = \boldsymbol{x}_j - \boldsymbol{x}_i \, \forall \, \boldsymbol{x}_j \in \boldsymbol{X}$
    for all $\boldsymbol{s}_k \in \boldsymbol{S}$ do
        $\boldsymbol{s}_t = \boldsymbol{s}_k - \boldsymbol{x}_i$
        if $-\boldsymbol{s}_t \in \text{cone}(\boldsymbol{X}_t)$ then
            $\boldsymbol{p}[k] = \text{true}$
        end if
        if all in $\boldsymbol{p}$ then
            return $\boldsymbol{p}$
        end if
    end for
end for
return $\boldsymbol{p}$
```

Algorithm C.3: Projection procedure for inside hull lying boundary point

```
Input: Hull Points $\boldsymbol{X}$, Labeled Boundary Point inside Hull $\boldsymbol{x}_j$
Output: Surrogate Point $\boldsymbol{x}_{ih}$
calculate center of gravity $\boldsymbol{\gamma} = \text{mean}(\boldsymbol{X})$
find support vector $\boldsymbol{v}_s = \boldsymbol{x}_j - \boldsymbol{\gamma}$
norm support vector $\boldsymbol{v}_{sn} = \boldsymbol{v}_s \left|\boldsymbol{v}_s\right|^{-1}$
for all $\boldsymbol{x}_i \in \boldsymbol{X}$ do
    calculate support $\rho(\boldsymbol{x}_i) = (\boldsymbol{x}_i - \boldsymbol{x}_j)\boldsymbol{v}_{sn}^\top$
end for
find point $\boldsymbol{x}_{i,max}$ with maximum support $\max(\boldsymbol{\rho})$
if $\boldsymbol{x}_{i,max}$ lies on $\boldsymbol{v}_{sn}$ then
    add slight random noise to support vector
    norm support vector
end if
project maximum support point $\boldsymbol{x}_{i,max,p} = \rho(\boldsymbol{x}_{i,max})\boldsymbol{v}_{sn}$
find closest point on hull by GJK $\boldsymbol{x}_{i,max,h}$
return $\boldsymbol{x}_{i,max,h}$
```

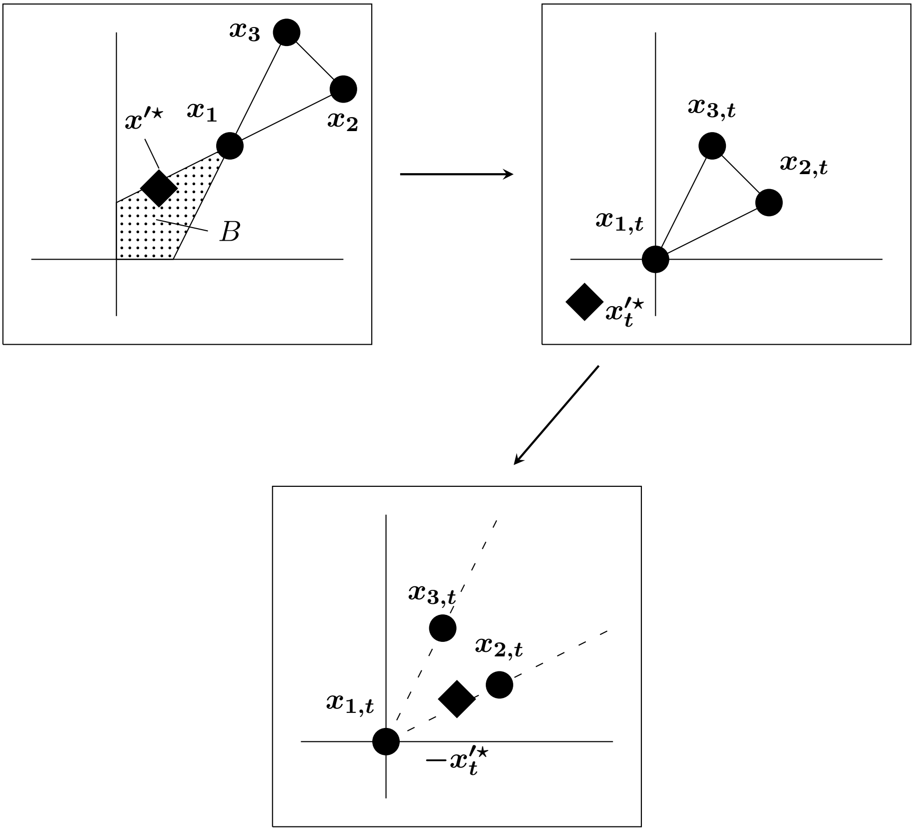


Figure C.1: Exemplary illustration of the exclusion area check for a test point $\boldsymbol{x'^{\star}}$ by a convex cone generated by $\boldsymbol{x_i}$. In this example $-\boldsymbol{x_t'^{\star}}$ is inside cone($\boldsymbol{X}$) and therefore $\boldsymbol{x'^{\star}}$ is part of exclusion area B.

D Mutual Information Test Design Comparison

Table D.1: Rating for different MI-based test design strategies and a non-adaptive space-filling test design by utilizing the CO output of the 7-dimensional diesel model

Method	AUC	$a + b$	Rating AUC	Rating Coefficients
Space Filling	494.2	-4.3	4	6
First 500 Candidates	507.5	-4.06	9	8
First 1000 Candidates	497.5	-4.37	6	5
First 1500 Candidates	503.2	-4.22	7	7
First 2000 Candidates	496.4	-4.47	5	4
500 Space-Filling Candidates	493.6	-4.76	3	3
1000 Space-Filling Candidates	491.3	-5.08	2	1
1500 Space-Filling Candidates	503.7	-4.04	8	9
2000 Space-Filling Candidates	489.4	-4.79	1	2

Table D.2: Rating for different MI-based test design strategies and a non-adaptive space-filling test design by utilizing the 9-dimensional radcosn model

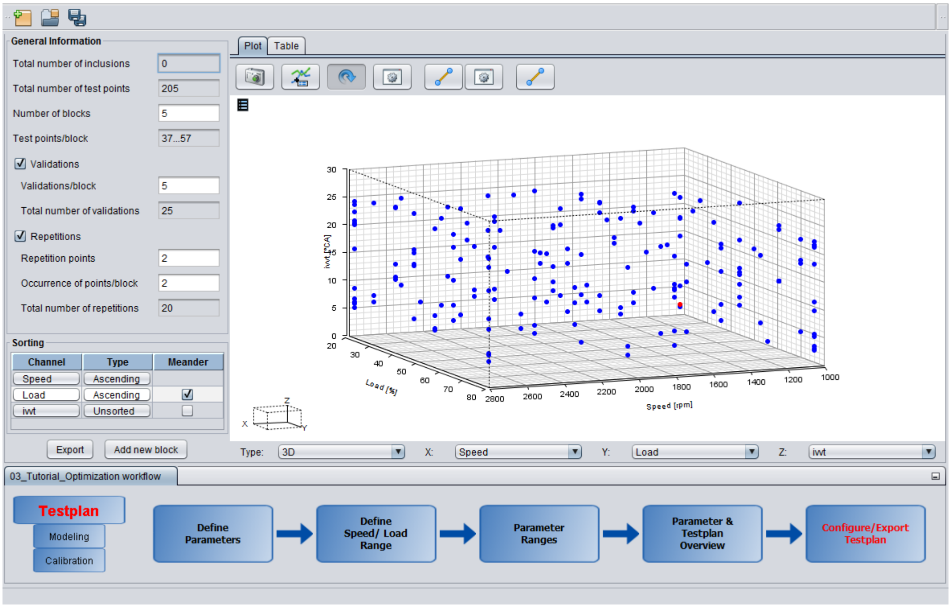
Method	AUC	$a + b$	Rating AUC	Rating Coefficients
Space Filling	720.8	-2.72	9	9
First 500 Candidates	630.5	-4.38	2	2
First 1000 Candidates	643.7	-4.04	6	6
First 1500 Candidates	639.6	-4.3	5	3
First 2000 Candidates	647.9	-4.05	8	5
500 Space-Filling Candidates	628.6	-4.39	1	1
1000 Space-Filling Candidates	645.8	-3.94	7	8
1500 Space-Filling Candidates	633.5	-4.17	3	4
2000 Space-Filling Candidates	638.2	-4.03	4	7

E IAV Kasai

The commercial software IAV Kasai [IAV20] is a tool that is designed to support the model-based engine calibration process. The calibration engineer is assisted during the three phases test design, model training, and optimization. For each phase, a workflow guides through the necessary steps to finally achieve an optimized actuator setting. Figure E.1 shows the software at the last step of the test plan workflow.

The relevant information to design the tests is the definition of the model inputs and the input ranges, in which information shall be gathered during the tests. Engine speed and load are defined by a user-specific grid, while all other inputs are continuously planned by default. To restrict the input domain, map-, table-, and inequation-based constraints are applicable as well as previously collected boundary measurements can be involved. With these definitions, a Sobol candidate set is created to gather test points from, which are finally selected by means of the maximin algorithm [JMY90]. Since the software is designed to be applicable for any test bench setup, there is no strictly required automation system to measure the test plan content. Once the measurement is collected, it is imported and the modeling workflow is started. A data analysis is part of this process with a model training in the final step. The four different model classes GPM, HILOMOT, polynomial model, and RBF can be selected. Several modeling methods can be applied to each output as well, whereupon an automatic statistically-based selection of the best performing model is executed.

Once a model is trained for each output, several methods to find the optimal settings can be chosen. A simple manual optimization by searching a minimum or maximum value of an output can be performed by an interactive single interaction plot. To set up a more complex optimization, a map-based and a reference-based optimization plug-in, a local optimization tool, and a trade-off calculation tool are available. The output of these tools are either optimized input maps or optimal input values, which need to be applied to the ECU software by the calibration engineer in a subsequent step.



General Information

Total number of inclusions: 0

Total number of test points: 205

Number of blocks: 5

Test points/block: 37...57

☑ Validations

Validations/block: 5

Total number of validations: 25

☑ Repetitions

Repetition points: 2

Occurrence of points/block: 2

Total number of repetitions: 20

Sorting

Channel	Type	Meander
Speed	Ascending	
Load	Ascending	☑
iwt	Unsorted	□

Type: 3D X: Speed Y: Load Z: iwt

Figure E.1: Exemplary screenshot of the test plan workflow of IAV Kasai. The last workflow step with already calculated test points and an exemplary visualization of the test point distribution is shown. The three main workflow steps are shown bottom left, the detailed test plan workflow is shown horizontally at the bottom.

F Test Bench Validation Model Error Trends

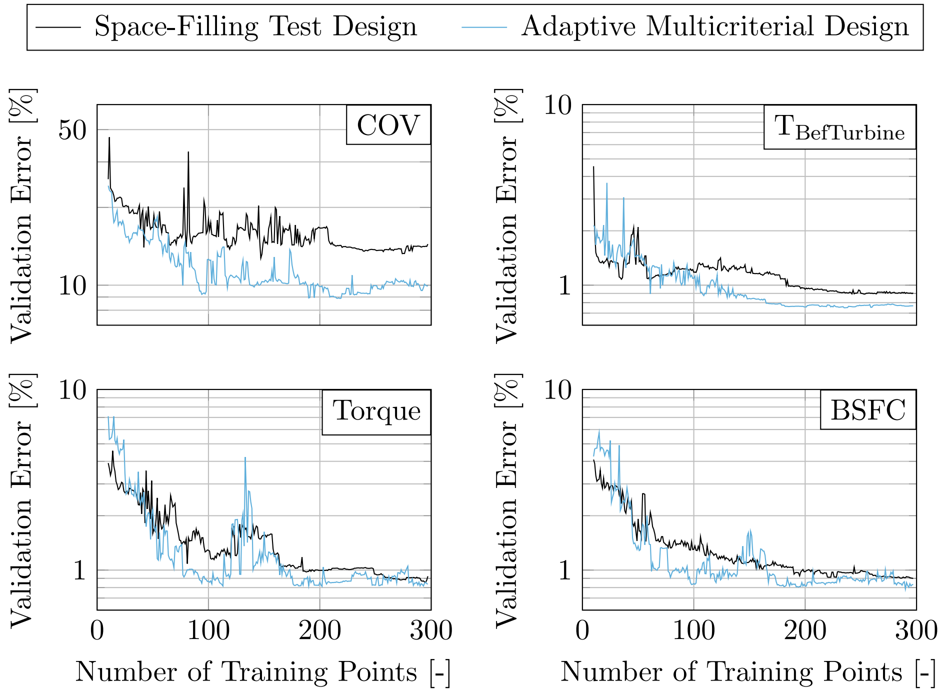


Figure F.1: Normed validation RMSE model quality trend given the amount of training points. From top left to bottom right the outputs running smoothness, exhaust temperature, engine torque and fuel consumptionare shown for the non-adaptive and the adaptive approach.

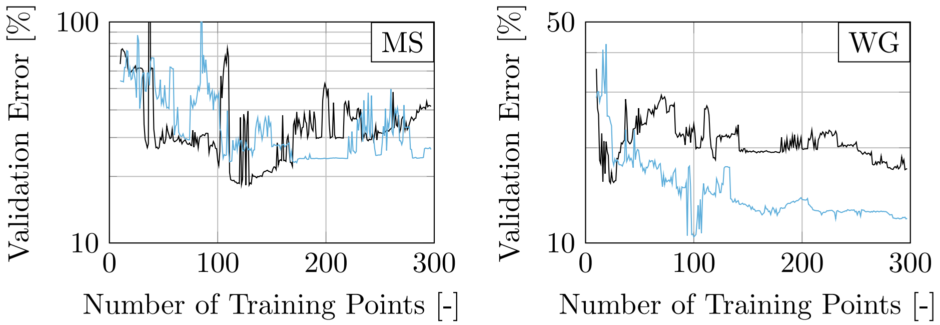


Figure F.2: Normed validation RMSE model quality trend given the amount of training points. The soot mass output (left) and the waste gate position output (right) are shown for the non-adaptive and the adaptive approach.

Bibliography

[AD20] A&D. *A&D ORION web page.* Last accessed on 21.10.2020. 2020. URL: `https://aanddtech.com/products/automated-calibration/`.

[Ami+19] A. Amini et al. "Spatial Uncertainty Sampling for End to End Control." In: *31st Conference on Neural Information Processing Systems (NIPS 2017).* 2019.

[AVL20] AVL. *AVL PUMA Open 2 web page.* Last accessed on 12.11.2020. 2020. URL: `https://www.avl.com/documents/10138/2095827/Product+brochure+Puma+Open+2`.

[Bar19] C.B. Barber. *QHull Webpage.* Last accessed on 21.10.2020. 2019. URL: `http://www.qhull.org/`.

[Bau+13] W. Baumann et al. "DoE for Series Production Calibration." In: *Design of Experiments (DoE) in Engine Development.* Vol. 6. Expert Verlag, 2013, pp. 125–135. ISBN: 9783816932178.

[BDH96] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. "The Quickhull Algorithm for Convex Hulls." In: *ACM Trans. Math. Softw.* 22.4 (Dec. 1996), pp. 469–483.

[Bel61] R. Bellman. *Adaptive Control Processes: A Guided Tour.* 1st ed. Princeton University Press, 1961. Chap. 5. ISBN: 978-0691625850.

[Ber12] B. Berger. "Modeling and Optimization for Stationary Base Engine Calibration." PhD thesis. Technical University of Munich, 2012.

[BF88] P. Bratley and B. L. Fox. "Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator." In: *ACM Trans. Math. Softw.* 14.1 (Mar. 1988), pp. 88–100. ISSN: 0098-3500.

[BGC07] A. Buzy, S. Grall, and P. Cotte. "A new generation of Online Adaptive Design of Experiments: Application to Emissions development and Full Load Calibration." In: *Design of Experiments (DoE) in Engine Development*. Vol. 3. Expert Verlag, 2007, pp. 76–90. ISBN: 9783816927204.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. 1st ed. Springer Science+Business Media, LLC, New York, 2006. Chap. 5, 6. ISBN: 978-0387-31073-2.

[Bol+13] A. Bolduan et al. "From Local Towards Global: Models for Diesel Engine Calibration." In: *Design of Experiments (DoE) in Engine Development*. Vol. 6. Expert Verlag, 2013, pp. 59–69. ISBN: 9783816932178.

[BP15] E. Burnaev and M. Panov. "Adaptive Design of Experiments Based on Gaussian Processes." In: *Statistical Learning and Data Sciences*. Ed. by A. Gammerman, V. Vovk, and H. Papadopoulos. Cham: Springer International Publishing, 2015, pp. 116–125.

[BR11] B. Berger and F. Rauscher. "Evaluation of Gaussian Processes for Black-Box Engine Modelling." In: *Design of Experiments (DoE) in Engine Development*. Vol. 5. Expert Verlag, 2011, pp. 376–397. ISBN: 9783816930747.

[BRL11] B. Berger, F. Rauscher, and B. Lohmann. "Analysing Gaussian Processes for Stationary Black-Box Combustion Engine Modelling." In: *IFAC Proceedings Volumes*. Vol. 44. 2011, pp. 10633–10640. ISBN: 978-3-902661-93-7.

[Cam97] S. Cameron. "Enhancing GJK: Computing Minimum and Penetration Distances between Convex Polyhedra." In: *Proc. Int. Conf. On Robotics and Automation* 4 (Feb. 1997).

[Cha+09] F. Chaudoye et al. "Modelling engine operating space for DoE calibration methods." In: *Design of Experiments (DoE) in Engine Development*. Vol. 4. Expert Verlag, 2009, pp. 107–121. ISBN: 9783816929376.

[CMP96] M. A. Clyde, P. Müller, and G. Parmigiani. *Exploring Expected Utility Surfaces by Markov Chains*. Tech. rep. 1996.

[Coc73] W. G. Cochran. “Experiments for Nonlinear Functions.” In: *Journal of the American Statistical Association* 68.344 (1973), pp. 771–781.

[CT17] R. W. Cottle and M. N. Thapa. *Linear and Nonlinear Optimization.* 1st ed. Springer New York, 2017. ISBN: 978-1-4939-7055-1.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* New York, NY, USA: Wiley-Interscience, 1991. Chap. 9. ISBN: 0-471-06259-6.

[CV95] K. Chaloner and I. Verdinelli. “Bayesian Experimental Design: A Review.” In: *Statistical Science* 10.3 (1995), pp. 273–304.

[DKR17] N. Didcock, N. Keuth, and A. Rainer. “Non-Convex Hulls for Engineering Applications.” In: *Automotive Data Analytics, Methods, DoE: Proceedings of the International Calibration Conference.* Vol. 8. Expert Verlag, 2017, pp. 309–320. ISBN: 9783816933816.

[Dwy+13] T. P. Dwyer et al. “DoE Framework for GDI Engine Mapping and Calibration Optimisation for CO2 and Particulate Number Emissions.” In: *Design of Experiments (DoE) in Engine Development.* Vol. 6. Expert Verlag, 2013, pp. 418–432. ISBN: 9783816932178.

[EK95] R. Eberhart and J. Kennedy. “A new optimizer using particle swarm theory.” In: *MHS’95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science.* Oct. 1995, pp. 39–43.

[EKP08] H. Eichlseder, M. Klüting, and W. F. Piock. *Grundlagen und Technologien des Ottomotors.* Springer-Verlag Wien, 2008, pp. 19–20. ISBN: 978-3-211-25774-6.

[ETA20a] ETAS. *ETAS ASCMO Webpage.* Last accessed on 21.10.2020. 2020. URL: `https://www.etas.com/en/products/ascmo.php`.

[ETA20b] ETAS. *ETAS INCA web page.* Last accessed on 12.11.2020. 2020. URL: `https://www.etas.com/de/portfolio/inca.php`.

[Fah+13] L. Fahrmeir et al. *Regression.* 1st ed. Springer-Verlag Berlin Heidelberg, 2013. Chap. 3, 4. ISBN: 978-3-642-34332-2.

[Fis26] R. A. Fisher. "048: The Arrangement of Field Experiments." In: *Journal of the Ministry of Agriculture of Great Britain*. Ed. by Great Britain. Ministry of Agriculture. Vol. 33. London: Ministry of Agriculture and Fisheries, 1926, pp. 83–94.

[For+03] T. Fortuna et al. "Optimierung von Motorsteuerungen am Motorenprüfstand." In: *Modellgestützte Steuerung, Regelung und Diagnose von Verbrennungsmotoren*. Vol. 1. Springer Berlin Heidelberg, 2003. Chap. 16, pp. 251–274. ISBN: 978-3-540-44286-8.

[Gen+15] Z. Geng et al. "Gaussian Process Based Modeling and Experimental Design for Sensor Calibration in Drifting Environments." In: *Sensors and Actuators B: Chemical* 216 (Apr. 2015).

[GJK88] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi. "A fast procedure for computing the distance between complex objects in three-dimensional space." In: *IEEE Journal on Robotics and Automation* 4.2 (Apr. 1988), pp. 193–203.

[GKS05] C. Guestrin, A. Krause, and A. P. Singh. "Near-optimal Sensor Placements in Gaussian Processes." In: *Proceedings of the 22Nd International Conference on Machine Learning*. ICML '05. Bonn, Germany: ACM, 2005, pp. 265–272.

[Gsc+01] K. Gschweitl et al. "Increasing the efficiency of model-based engine applications through the use of CAMEO online DoE toolbox." In: *ATZ worldwide* 103.7 (July 2001), pp. 17–20.

[Har+16] B. Hartmann et al. "Online-methods for engine test bed measurements considering engine limits." In: *16. Internationales Stuttgarter Symposium*. Ed. by M. Bargende, H.-C. Reuss, and J. Wiedemann. Wiesbaden: Springer Fachmedien Wiesbaden, 2016, pp. 1251–1264. ISBN: 978-3-658-13255-2.

[Har14] B. Hartmann. "Lokale Modellnetze zur Identifikation und Versuchsplanung nichtlinearer Systeme." PhD thesis. University of Siegen, 2014.

[Har15] F.E. Harrell. *Regression Modeling Strategies*. Springer International Publishing, 2015. Chap. 5. ISBN: 978-3-319-19425-7.

[HD16] K. Hoag and B. Dondlinger. *Vehicular Engine Design.* 2nd ed. Springer Verlag Wien, 2016. Chap. 4. ISBN: 978-3-7091-1859-7.

[Hei+13] A. Heinstein et al. "Direkteinspritzsysteme für ottomotoren." In: *MTZ - Motortechnische Zeitschrift* 74.3 (Mar. 2013), pp. 226–231. ISSN: 0024-8525.

[Hey88] J.B. Heywood. *Internal Combustion Engine Fundamentals.* 1st ed. McGraw-Hill, 1988, pp. 417, 376–378, 51. ISBN: 9780070286375.

[HJ01] R. J. Howlett and L. C. Jain. *Radial Basis Function Networks 2.* 1st ed. Physica-Verlag Heidelberg New York, 2001. Chap. 1. ISBN: 978-3-7908-2483-4.

[HJ11] C. Hametner and S. Jakubek. "Combustion engine modelling using an evolving local model network." In: *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011).* June 2011, pp. 2802–2807.

[HK70] A. E. Hoerl and R. W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." In: *Technometrics* 12.1 (1970), pp. 55–67. ISSN: 00401706. URL: `http://www.jstor.org/stable/1267351`.

[HN13] B. Hartmann and O. Nelles. "Adaptive Test Planning for the Calibration of Combustion Engines - Methodology." In: *Design of Experiments (DoE) in Engine Development.* Vol. 6. Expert Verlag, 2013, pp. 1–16. ISBN: 9783816932178.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction.* 2nd ed. Springer, 2009. Chap. 2, 3, 7. ISBN: 978-0387848570.

[IAV20] IAV. *IAV Kasai Webpage.* Last accessed on 21.10.2020. 2020. URL: `https://www.iav.com/en/services/products/tools/iav-kasai`.

[IS20] H. Ihme-Schramm and A. Schramm. "Integration der virtuellen Motorapplikation in die reale Entwicklungsumgebung." In: *MTZ - Motortechnische Zeitschrift* 81.7–8 (July 2020), pp. 54–58. ISSN: 0024-8525.

[Jam+13] G. James et al. *An Introduction to Statistical Learning.* 1st ed. Springer New York, 2013. Chap. 6. ISBN: 978-1-4614-7137-0.

[JH08] V. R. Joseph and Y. Hung. "Orthogonal-Maximin Latin Hypercube Designs." In: *Statistica Sinica* 18.1 (2008), pp. 171–186.

[JMY90] M. E. Johnson, L. M. Moore, and D. Ylvisaker. "Minimax and maximin distance designs." In: *Journal of Statistical Planning and Inference* 26.2 (1990), pp. 131–148.

[Joh87] D. W. Johnson. "The Optimization of Robot Motion in the Presence of Obstacles." PhD thesis. Ann Arbor, MI, USA, 1987.

[Kay93] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993. Chap. 3. ISBN: 0-13-345711-7.

[Kie14] N. Kieft. "Evaluation of Different Design Space Description Methods for Analysing Combustion Engine Operation Limits." PhD thesis. Leiden University, 2014.

[Kle+13] P. Klein et al. "Adaptive Test Planning for the Calibration of Combustion Engines - Application." In: *Design of Experiments (DoE) in Engine Development.* Vol. 6. Expert Verlag, 2013, pp. 17–30. ISBN: 9783816932178.

[Klö09] F. Klöpper. "Entwicklung und Einsatz modellgestützter Online-Methoden zur Parameteroptimierungvon Verbrennungsmotoren am Prüfstand." PhD thesis. Technical University of Munich, 2009.

[KMK17] S. Kiga, K. Moteki, and S. Kojima. "The New Nissan VC-Turbo with Variable Compression Ratio." In: *MTZ worldwide* 78.11 (Nov. 2017), pp. 42–48. ISSN: 2192-9114.

[Knö+03] K. Knödler et al. "Modellbasierte Online-Optimierung moderner Verbrennungsmotoren Teil 2: Grenzen des fahrbaren Suchraums." In: *MTZ - Motortechnische Zeitschrift* 64.6 (June 2003), pp. 520–526. ISSN: 2192-8843.

[Kow17] M. Kowalczyk. "Online-Methoden zur effizienten Vermessung von statischen und dynamischen Verbrennungsmotormodellen." PhD thesis. Technical University of Darmstadt, 2017.

[Kra17] O. Kramer. *Genetic Algorithm Essentials.* 1st ed. Springer International Publishing, 2017. ISBN: 978-3-319-52156-5.

[KSR06] M. Knaak, S. Schaum, and K. Röpke. "Fast measurement by integrating model-based methods and advanced test-cell automation." In: *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control.* Aug. 2006, pp. 2054–2059.

[KUS07] T. Kruse, H. Ulmer, and U. Schulmeister. "Use of Advanced Modelling and Optimization for Diesel- and Gasoline Engine Calibration." In: *Design of Experiments (DoE) in Engine Development.* Vol. 3. Expert Verlag, 2007, pp. 91–102. ISBN: 9783816927204.

[Lin56] D. V. Lindley. "On a Measure of the Information Provided by an Experiment." In: *The Annals of Mathematical Statistics* 27.4 (1956), pp. 986–1005.

[Mit74] T. J. Mitchell. "An Algorithm for the Construction of D-Optimal Experimental Designs." In: *Technometrics.* Vol. 16. 1974, pp. 203–210.

[MM95] M. D. Morris and T. J. Mitchell. "Exploratory Designs for Computational Experiments." In: *Journal of Statistical Planning and Inference* 43 (Feb. 1995), pp. 381–402.

[Mon08] D.C. Montgomery. *Design and Analysis of Experiments.* 8th ed. Student solutions manual. John Wiley & Sons, 2008, pp. 280–284. ISBN: 978-1118-14692-7.

[MTI03] E. Martini, H. Voßand S. Töpfer, and R. Isermann. "Effiziente Motorapplikation mit lokal linearen neuronalen Netzen." In: *MTZ - Motortechnische Zeitschrift* 64.5 (May 2003), pp. 406–413. ISSN: 2192-8843.

[Mur+15] Y. Murata et al. "Application of Model Based Calibration to Mass Production Diesel Engine Development for Indian Market." In: *Design of Experiments (DoE) in Powertrain Development.* Vol. 7. Expert Verlag, 2015, pp. 77–88. ISBN: 9783816933168.

[Nea96] R.M. Neal. *Bayesian Learning for Neural Networks.* Lecture Notes in Statistics. New York: Springer New York, 1996, pp. 29–42.

[Nel06] O. Nelles. "Neuronale Netze: Eine Übersicht." In: *Informationsfusion in der Mess- und Sensortechnik*. Ed. by J. Beyerer, F. Puente León, and K.-D. Sommer. Vol. 1. Universitätsverlag Karlsruhe, Karlsruhe, 2006, pp. 35–112, 238, 296. ISBN: 3-86644-053-7.

[Nel20] O. Nelles. *Nonlinear System Identification*. 2nd ed. Springer-Verlag, 2020. Chap. 1, 7, 10, 13. ISBN: 978-3-030-47438-6.

[Nis20] Nissan. *Nissan webpage*. Last accessed on 20.07.2020. 2020. URL: `https://www.nissan-global.com/EN/TECHNOLOGY/OVERVIEW/vc_turbo_engine.html`.

[NSI96] O. Nelles, S. Sinsel, and R. Isermann. "Local basis function networks for identification of a turbocharger." In: *UKACC International Conference on Control '96 (Conf. Publ. No. 427)*. Vol. 1. Sept. 1996, pp. 7–12.

[Pol+03] J. Poland et al. "Modellbasierte Online-Optimierung moderner Verbrennungsmotoren Teil 1: Aktives Lernen." In: *MTZ - Motortechnische Zeitschrift* 64.5 (May 2003), pp. 432–437. ISSN: 2192-8843.

[Pol02] J. Poland. "Modellgestützte und Evolutionäre Optimierungsverfahren für die Motorenentwicklung." PhD thesis. Eberhard-Karls-University Tübingen, 2002.

[RA05] P. Renninger and M. Aleksandrov. "Rapid Hull Determination: a new method to determine the design space for model based approaches." In: *Design of Experiments (DoE) in Engine Development*. Vol. 2. Expert Verlag, 2005, pp. 14–29. ISBN: 978-3816924982.

[Rai15] B. Raidt. "A strategy to employ criteria for online location selection when using Gaussian Processes." In: *Design of Experiments (DoE) in Powertrain Development*. Vol. 7. Expert Verlag, June 2015, pp. 1–14. ISBN: 9783816933168.

[Ran+13] J. v. Rango et al. "Comparison of Different Approaches for Global Modeling of Combustion Engines." In: *Design of Experiments (DoE) in Engine Development*. Vol. 6. Expert Verlag, 2013, pp. 70–91. ISBN: 9783816932178.

[Röp+07] K. Röpke et al. “Rapid Measurement: Grundbedatung eines Verbrennungsmotors innerhalb eines Tages?” In: *MTZ - Motortechnische Zeitschrift* 68.4 (Apr. 2007), pp. 276–282. ISSN: 2192-8843.

[RRC19] G. Rebala, A. Ravi, and S. Churiwala. *An Introduction to Machine Learning*. Springer International Publishing, 2019. Chap. 8. ISBN: 978-3-030-15729-6.

[RW06] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, Jan. 2006.

[Rya+15] E. G. Ryan et al. “A Review of Modern Computational Algorithms for Bayesian Optimal Design.” In: *International Statistical Review* 84 (June 2015).

[Sac+89] J. Sacks et al. “Design and Analysis of Computer Experiments.” In: *Statistical Science* 4.4 (1989), pp. 409–423.

[Sam09] D. Sampson. “Use of Model-Based Calibration in the Test Cell.” In: *Design of Experiments (DoE) in Engine Development*. Vol. 4. Expert Verlag, 2009, pp. 23–30. ISBN: 9783816929376.

[SBH10] K. Siebertz, D. Bebber, and T. Hochkirchen. *Statistische Versuchsplanung – Design of Experiments (DoE)*. 1st ed. Springer-Verlag Berlin Heidelberg, 2010, pp. 53, 54, 168–173. ISBN: 978-3-642-05492-1.

[Sch+15] J. Schreiter et al. “Safe Exploration for Active Learning with Gaussian Processes.” In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by A. Bifet et al. Springer International Publishing, 2015, pp. 133–149.

[Sch+17] M. Schillinger et al. “Modern Online DoE Methods for Calibration - Constraint Modeling, Continuous Boundary Estimation, and Active Learning.” In: *Automotive Data Analytics, Methods, DoE: Proceedings of the International Calibration Conference*. Vol. 8. Expert Verlag, 2017, pp. 321–335. ISBN: 9783816933816.

[Sch+18] M. Schillinger et al. "Safe Active Learning of a High Pressure Fuel Supply System." In: *Proceedings of The 9th EUROSIM Congress on Modelling and Simulation, EUROSIM 2016, The 57th SIMS Conference on Simulation and Modelling SIMS 2016*. Ed. by E. Juuso, E. Dahlquist, and K. Leiviskä. Linköping University Electronic Press, Linköpings Universitet, Dec. 2018, pp. 286–292. ISBN: 978-91-7685-399-3.

[Sch96] R. Schwabe. *Optimum Designs for Multi-Factor Models*. 8th ed. Lecture Notes in Statistics. Springer New York, 1996, pp. 8–10. ISBN: 978-1-4612-4038-9.

[Set09] B. Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison, 2009.

[SI09] A. Schreiber and R. Isermann. "Methods for global dynamic measurement and modelling of combustion engines." In: *Design of Experiments (DoE) in Engine Development*. Vol. 4. Expert Verlag, 2009, pp. 293–305. ISBN: 9783816929376.

[Sob67] I. M. Sobol. "On the distribution of points in a cube and the approximate evaluation of integrals." In: *USSR Computational Mathematics and Mathematical Physics*. Vol. 7. 4. 1967, pp. 86–112.

[SOS92] H. S. Seung, M. Opper, and H. Sompolinsky. "Query by Committee." In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: ACM, 1992, pp. 287–294. ISBN: 0-89791-497-X.

[SS17a] S. Schaum and N. Sandmeier. *Verfahren und Vorrichtung zur Ermittlung eines Modells eines Technischen Systems*. Assignee IAV GMBH Ingenieurgesellschaft Auto & Verkehr, patent number DE102016106976A1. Aug. 2017.

[Sta+11] M. Stadlbauer et al. "Analytic Model based Design of Experiments." In: *Design of Experiments (DoE) in Engine Development*. Vol. 5. Expert Verlag, 2011, pp. 13–23. ISBN: 9783816930747.

[Sta+13] M. Stadlbauer et al. “Online Design of Experiments in the Relevant Output Range.” In: *Design of Experiments (DoE) in Engine Development*. Vol. 6. Expert Verlag, 2013, pp. 43–58. ISBN: 9783816932178.

[Sta+14] M. Stadlbauer et al. *Methode zur Ermittlung eines Modells einer Ausgangsgröße eines technischen Systems*. Assignee AVL LIST GMBH, patent number A50347/2013. Dec. 2014.

[Sui+15] Y. Sui et al. “Safe Exploration for Optimization with Gaussian Processes.” In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning*. Vol. 37. ICML ’15. Lille, France: JMLR.org, 2015, pp. 997–1005.

[Sun+07] A. Sung et al. “Modellbasierte Online-Optimierung in der Simulation und am Motorenprüfstand.” In: *MTZ - Motortechnische Zeitschrift* 68.1 (Jan. 2007), pp. 42–48. ISSN: 2192-8843.

[SV17] S. Scheidel and M.-S. Vogels. “Model-based iterative DoE in highly constrained spaces.” In: *Automotive Data Analytics, Methods, DoE: Proceedings of the International Calibration Conference*. Vol. 8. Expert Verlag, 2017, pp. 336–348. ISBN: 9783816933816.

[SW00] P. Sebastiani and H. P. Wynn. “Maximum Entropy Sampling and Optimal Bayesian Experimental Design.” In: *Journal of the Royal Statistical Society Series B* 62 (Feb. 2000), pp. 145–157.

[SW87] M. C. Shewry and H. P. Wynn. “Maximum entropy sampling.” In: *Journal of Applied Statistics* 14.2 (1987), pp. 165–170.

[SWN18] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. 2nd ed. Springer New York, 2018. Chap. 5,6. ISBN: 978-1-4939-8845-7.

[The+16] S. Thewes et al. “Efficient In-Vehicle Calibration by the Usage of Automation and Enhanced Online DoE Approaches.” In: *Simulation and Testing for Vehicle Technology*. Vol. 21. Springer International Publishing, 2016, pp. 299–316. ISBN: 978-3-319-32345-9.

[Tie15] N. Tietze. “Model-based Calibration of Engine Control Units Using Gaussian Process Regression.” PhD thesis. Technical University of Darmstadt, 2015.

[War95] T. T. Warnock. “Computational Investigations of Low-Discrepancy Point Sets II.” In: Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing. Springer New York, 1995, pp. 354–361. ISBN: 978-0-387-94577-4.

[Wel84] W. J. Welch. “Computer-Aided Design of Experiments for Response Estimation.” In: *Technometrics*. Vol. 26. 1984, pp. 217–224.

[WKO15] S. Watanabe, J. Kako, and A. Ohata. “Benchmark Problem for Near Boundary Operation Control for Automotive Engine.” In: *Design of Experiments (DoE) in Powertrain Development*. Vol. 7. Expert Verlag, 2015, pp. 30–40. ISBN: 9783816933168.

[WR96] C. K. I. Williams and C. E. Rasmussen. “Gaussian Processes for Regression.” In: *Advances in Neural Information Processing Systems 8*. MIT press, 1996, pp. 514–520.

[Xie16] Y. Xie. “Optimal Steady-State Base-Calibration of Model Based ECU-Functions.” In: *Simulation and Testing for Vehicle Technology*. Ed. by C. Gühmann, J. Riese, and K. von Rüden. Cham: Springer International Publishing, 2016, pp. 245–265.

[Yos+11] S. Yoshida et al. “Rapid Boundary Detection for Model Based Diesel Engine Calibration.” In: *Design of Experiments (DoE) in Engine Development*. Vol. 5. Expert Verlag, 2011, pp. 36–55. ISBN: 9783816930747.

[Zag14] S. Zaglauer. “Methode zur multikriteriellen Optimierung des Motorverhaltens anhand physikalisch motivierter Modelle.” PhD thesis. Friedrich-Alexander University Erlangen-Nürnberg, 2014.

[ZC09] Y. Zheng and C.-M. Chew. “Distance Between a Point and a Convex Cone in n-Dimensional Space: Computation and Applications.” In: *IEEE Transactions on Robotics* 25.6 (Dec. 2009), pp. 1397–1412.

[ZD11] S. Zaglauer and M. Deflorian. “Bayesian D-optimal Design.” In: *Design of Experiments (DoE) in Engine Development.* Vol. 5. Expert Verlag, 2011, pp. 1–13. ISBN: 9783816930747.

[Zha+09] Q. Zhang et al. *Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition.* Tech. rep. Apr. 2009.

[ZKN12] Z. Zainudin, S. Kodagoda, and L. Nguyen. “Mutual information based data selection in Gaussian processes for people tracking.” In: *Proceedings of Australasian Conference on Robotics and Automation.* Dec. 2012.

Publications

[San+20] N. Sandmeier et al. "High-Dimensional Adaptive Test Design Including Boundary Search." In: *SAE International Journal of Engines.* Vol. 13. Feb. 2020.

[SR15a] N. Sandmeier and K. Röpke. "Efficient online test plan calculation with integrated boundary detection for modern model based engine calibration." In: *Design of Experiments (DoE) in Powertrain Development.* Vol. 7. Expert Verlag, June 2015, pp. 15–29. ISBN: 9783816933168.

[SR15b] N. Sandmeier and K. Röpke. "Improving the usability and time effort of modern engine calibration tasks by means of an online, model based approach." In: *6th International Symposium on Development Methodology.* Nov. 2015, pp. 127–138. ISBN: 9783981697117.

[SS17b] S. Schaum and N. Sandmeier. *Verfahren und Vorrichtung zur Ermittlung eines Modells eines Technischen Systems.* Assignee IAV GMBH Ingenieurgesellschaft Auto & Verkehr, patent number DE102016106976A1. Aug. 2017.

Supervised Students' Theses

[Lep17] K. Lepke. "Potentialuntersuchung bezüglich der Einbindung von Rapid Measurement Methoden in die Online DoE." Master thesis. Hamburg University of Applied Sciences, Jan. 2017.

[Mel15] A. Mellin. "Optimierung von automatisierten, iterativen Versuchsplanerstellungsprozessen zur Vermessung von Verbrennungskraftmaschinen am Motorenprüfstand nach den Regeln der statistischen Versuchsplanung." Master thesis. Hamburg University of Applied Sciences, May 2015.

[Mew17] G. Mewes. "Entwicklung und Validierung von echtzeitfähigen Methoden für die automatisierte Vermessung von Verbrennungsmotoren unter der Berücksichtigung beschränkter Eingangsräume." Master thesis. Hamburg University of Applied Sciences, Sept. 2017.

Advances in Automation Engineering
Hrsg.: Prof. Dr.-Ing. Clemens Gühmann
ISSN 2509-8950 (print)
ISSN 2509-8969 (online)

1. Nowoisky, Sebastian: Verfahren zur Identifikation nichtlinearer dynamischer Getriebemodelle. - 2016. - VIII, 224 S.
ISBN 978-3-7983-2854-9 (print) 15,00 EUR
ISBN 978-3-7983-2855-6 (online)
DOI 10.14279/depositonce-5420

2. Huang, Hua: Model-based calibration of automated transmissions. - 2016. - XXIV, 134 S.
ISBN 978-3-7983-2858-7 (print) 14,00 EUR
ISBN 978-3-7983-2859-4 (online)
DOI 10.14279/depositonce-5461

3. Röper, Jan: Entwicklung eines virtuellen Getriebeprüfstands. - 2017. - xxvi, 133 S.
ISBN 978-3-7983-2951-5 (print) 14,00 EUR
ISBN 978-3-7983-2952-2 (online)
DOI 10.14279/depositonce-6073

4. Funck, Jürgen Helmut: Synchronous data acquisition with wireless sensor networks. - 2018. - xix, 327 S.
ISBN 978-3-7983-2980-5 (print) 19,50 EUR
ISBN 978-3-7983-2981-2 (online)
DOI 10.14279/depositonce-6716

5. Kiffe, Axel: Echtzeitsimulation leistungselektronischer Schaltungen für die Hardware-in-the-Loop-Simulation. - 2018. - x, 212 S.
ISBN 978-3-7983-3013-9 (print) 14,00 EUR
ISBN 978-3-7983-3014-6 (online)
DOI 10.14279/depositonce-7227

6. Lück, Rudolf: Überwachung hybrider Schrägkugellager in Luftfahrttriebwerken. - 2018. - XXIII, 169 S.
ISBN 978-3-7983-3021-4 (print) 18,50 EUR
ISBN 978-3-7983-3022-1 (online)
DOI 10.14279/depositonce-7283

7. Mokhtari, Noushin: Überwachung hydrodynamischer Gleitlager basierend auf der Körperschallanalyse. - 2020. - XXI, 167 S.
ISBN 978-3-7983-3183-9 (print) 18,50 EUR
ISBN 978-3-7983-3184-6 (online)
DOI 10.14279/depositonce-10642

8. Strommenger, Daniel: Verschleißprognose zur zuverlässigkeitsorientierten Regelung für trockene Reibkupplungen. - 2021. - xii, 236 S.
ISBN 978-3-7983-3196-9 (print) 19,50 EUR
ISBN 978-3-7983-3197-6 (online)
DOI 10.14279/depositonce-11299

9. Brunken, Hauke: Stereo vision-based road condition monitoring. - 2021. - xxiii, 159 S.
ISBN 978-3-7983-3205-8 (print) 15,00 EUR
ISBN 978-3-7983-3206-5 (online)
DOI 10.14279/depositonce-11487

Universitätsverlag der TU Berlin

ELEKTRONISCHE MESS- UND DIAGNOSETECHNIK

Optimization of adaptive test design methods for the determination of steady-state data-driven models in terms of combustion engine calibration

This thesis deals with the development of a model-based adaptive test design strategy with a focus on steady-state combustion engine calibration. The first research topic investigates the question how to handle limits in the input domain during an adaptive test design procedure. The second area of scope aims at identifying the test design method providing the best model quality improvement in terms of overall model prediction error. To consider restricted areas in the input domain, a convex hull-based solution involving a convex cone algorithm is developed, the outcome of which serves as a boundary model for a test point search. A solution is derived to enable the application of the boundary model to high-dimensional problems without calculating the exact convex hull and cones. Furthermore, different data-driven engine modeling methods are compared, resulting in the Gaussian process model as the most suitable one for a model-based calibration. To determine an appropriate test design method for a Gaussian process model application, two new strategies are developed and compared to state-of-the-art methods.

ISBN 978-3-7983-3247-8 (print)
ISBN 978-3-7983-3248-5 (online)

ISBN 978-3-7983-3247-8

https://verlag.tu-berlin.de