h_da
hochschule darmstadt
fachbereich mathematik
und naturwissenschaften

member of
EUT+
EUROPEAN UNIVERSITY OF TECHNOLOGY

Exploring and Evaluating Adaptive Design-of-Experiment Approaches for Enhanced Drivability Optimisation in Vehicle Manoeuvres

Master's thesis submitted in partial fulfilment of the requirements for the academic degree
Master of Science (M. Sc.) in Data Science

Submitted by

Tilman Seeßelberg

Matriculation Number: 1126856

Hochschule Darmstadt

Prof. Dr. Andreas Thümmel (Supervisor)
Prof. Dr. Timo Schürg (Co-Supervisor)

Mercedes-Benz Group AG

Dr.-Ing. Philipp Skarke
M.Sc Sebastian Körner

Issue date: 06.06.2025

Submission date: 05.12.2025

Abstract

Drivability optimisation in automotive context traditionally relies on static Design of Experiments and manual expert iteration, resulting in a high measurement effort and limited scalability. This thesis investigates whether adaptive Design of Experiments (aDoE) can improve the efficiency of this process while remaining compatible with existing industrial workflows. To this end, an integration-focused aDoE framework is developed that augments the current test-rig infrastructure with asynchronous sampling and surrogate-model updates using only minor architectural modifications.

The proposed approach is first evaluated in a simulative study based on the Hartmann-6 benchmark, comparing several adaptive sampling strategies against static baselines. The results indicate that exploitation-oriented strategies such as Expected Improvement and Upper Confidence Bound can reduce the required sample budget for identifying high-performing regions, albeit at the cost of reduced global model accuracy. A test-rig experiment applying an Expected Improvement-based strategy to the Change-of-Mind Engine Start manoeuvre demonstrates the practical feasibility of the framework. Subjective validation suggests that solutions of quality comparable to the standard workflow can be obtained with less than 60 % of the measurements, while also reducing the number of invalid manoeuvres through adaptive sampling.

While these findings indicate that aDoE can make drivability optimisation more efficient and robust, they also reveal important limitations regarding global model fidelity, uncertainty calibration, and the lack of comprehensive test-rig validation. The results therefore provide an encouraging but ultimately tentative step towards integrating aDoE into industrial calibration workflows and highlight directions for further methodological and infrastructural improvements.

Kurzzusammenfassung

Die Drivability-Optimierung in der automobilen Applikation stützt sich traditionell auf statische Design-of-Experiments-Methoden und manuelle iteratives Eingreifen durch Experten, was zu einem hohen Messaufwand und begrenzter Skalierbarkeit führt. Diese Arbeit untersucht, ob adaptive Design of Experiments (aDoE) die Effizienz dieses Prozesses steigern kann, ohne die Kompatibilität mit bestehenden industriellen Workflows zu beeinträchtigen. Hierzu wird ein integrationsorientiertes aDoE-Framework entwickelt, das die bestehende Prüfstandsinfrastruktur durch asynchrones Sampling und Surrogatmodell-Updates mit wenigen Anpassungen erweitert.

Der vorgeschlagene Ansatz wird zunächst in einer simulativen Studie auf Basis des Hartmann-6-Benchmarks evaluiert, in der mehrere adaptive Sampling-Strategien mit einer statischen Baseline verglichen werden. Die Ergebnisse zeigen, dass exploitation-orientierte Strategien wie Expected Improvement und Upper Confidence Bound die Anzahl der notwendigen Messungen zur Identifikation optimaler Regionen reduzieren können, allerdings auf Kosten einer geringeren globalen Modellgenauigkeit. Ein Prüfstandsexperiment, bei dem unter anderem eine auf Expected Improvement basierende Strategie auf das Change-of-Mind Engine Start-Manöver angewendet wurde, demonstriert die praktische Umsetzbarkeit des Frameworks. Subjektive Validierung legt nahe, dass Lösungen von mit dem Standard-Workflow vergleichbarer Qualität mit weniger als 60 % der Messungen erzielt werden können, während durch adaptives Sampling zugleich die Anzahl ungültiger Manöver reduziert wird.

Obgleich diese Befunde darauf hinweisen, dass aDoE die Drivability-Optimierung effizienter und robuster gestalten kann, zeigen sie zugleich wesentliche Einschränkungen hinsichtlich globaler Modelqualität, Unsicherheitskalibrierung und der fehlenden umfassenden Prüfstandsvalidierung auf. Die Ergebnisse stellen daher einen ermutigenden, letztlich jedoch vorläufigen Schritt in Richtung einer Integration von aDoE in industrielle Applikationsworkflows dar und markieren Ansatzpunkte für weitere methodische und infrastrukturelle Weiterentwicklungen.

Acknowledgements

I would like to express my sincere gratitude to Mercedes-Benz Group AG for giving me the opportunity to complete this thesis within their organisation. I am particularly grateful to Philipp Skarke and Sebastian Körner for giving me this opportunity, for trusting me, and for creating such a supportive environment. The experience I gained at Mercedes-Benz was invaluable, and the guidance and insights I received were consistently constructive, greatly enriching my understanding of the subject matter.

I would also like to thank Prof. Dr. Andreas Thümmel for our many technical discussions, his thoughtful ideas and his support. His expertise and constructive feedback were instrumental in helping me identify a clear, rigorous pathway to completing this research.

Disclaimer

I hereby declare that I have written this thesis independently and that I have not used any sources or aids other than those listed in the bibliography and explicitly mentioned in this disclaimer. All passages that are reproduced verbatim or in substance from published or unpublished sources are clearly identified as such. All drawings and figures in this thesis were created by me or are accompanied by an appropriate source reference. This thesis has not been submitted, in whole or in part, in the same or a similar form to any other examination authority. No AI tools were used to generate original scientific content, reasoning, or results in this thesis. *DeepL Write Pro*[1] was used solely for grammar and phrasing improvements. Additionally, *Consensus*[2] was used to support the literature review by helping to identify relevant research papers. Furthermore, *GitHub Copilot*[3] was used to assist in coding and refactoring the adaptive Design-of-Experiments (ADoE) framework implemented as part of this thesis. All intellectual contributions, interpretations, and conclusions presented in this work are entirely my own.

Place, Date: **Signature:**

Stuttgart, 5th December 2025 Tilman Seeßelberg

[1] `https://www.deepl.com/de/products/write` (visited on 17.11.2025)

[2] `https://consensus.app` (visited on 17.11.2025)

[3] `https://github.com/features/copilot` (visited on 17.11.2025)

Contents

List of Figures

List of Tables

List of Abbreviations

R^2	Coefficient of Determination *(p. 12, 15, 21, 50)*
aDoE	Adaptive Design of Experiment *(p. vi, 1, 2, 4–6, 8, 9, 16, 18, 19, 24, 25, 28–32, 35, 37, 40, 45–47, 49–60, 62, 63)*
AF	Acquisition Function *(p. 12, 18–21, 28, 30, 31, 33, 37, 40, 49–51, 53)*
AL	Active Learning *(p. 19)*
ANOVA	Analysis of Variance *(p. 26, 27, 34, 35, 37–40, 42, 43)*
ARD	Automatic Relevance Detection *(p. 10, 30, 31)*
CDF	cumulative distribution function *(p. 20)*
COM-ES	Change-of-Mind Engine Start *(p. 2, 6–8, 45, 49, 50, 57, 58, 62)*
DoE	Design of Experiment *(p. 1, 2, 4–6, 15, 16, 18, 19, 28, 31, 45–47, 49–55, 58, 62)*
ECU	Electronic Control Unit *(p. 1, 4, 46, 52)*
EI	Expected Improvement *(p. 20–22, 28, 33, 37, 50–54, 57, 58, 62)*
FFD	Full Factorial Designs *(p. 15, 16)*
GPR	Gaussian Process Regressor *(p. 5, 9, 10, 12, 14, 19–21, 28, 30, 33, 37, 48, 53, 56–59, 62)*
HiLoMoT	Hierarchical Local Model Tree *(p. 28, 31, 60, 61)*
HPO	Hyper Parameter Optimisation *(p. 12, 31, 53, 58, 59)*
HV	Hypervolume *(p. 24, 53)*
LHS	Latin Hypercube Sampling *(p. 18, 31–33, 48, 49)*
LMN	Local Model Network *(p. 9)*
MB	Mercedes-Benz Group AG *(p. 1, 5, 30)*
ML	Machine Learning *(p. 1, 9, 10)*
MOO	multi-objective Optimisation *(p. 4, 5, 22–24, 28, 30, 50)*
NSGA-II	Non-dominated Sorting Genetic Algorithm II *(p. 5, 23, 24, 51, 53)*
PDF	probability density function *(p. 20)*
PF	Pareto Front *(p. 5, 22–24, 52, 53, 57)*
R2R	Road-to-Rig *(p. 45, 47)*
RBF	Radial Basis Function *(p. 11, 25)*
RF	Random Forest *(p. 9, 12, 14)*
RMSE	Root Mean Square Error *(p. 12, 15, 26, 33–37, 39–42, 56–58)*
RQ	Rational Quadratic *(p. 11, 30)*
SM	Surrogate Model *(p. v, 1, 2, 4–6, 8, 9, 11, 14, 16, 18–21, 23, 24, 28–31, 34, 37, 38, 42, 47, 48, 56, 58–63)*
SMBO	Sequential model-based Global Optimisation *(p. 19)*
SSB	sum of squares between *(p. 26)*
SST	total Sum of Squares *(p. 26)*
SSW	sum of squares within *(p. 26)*
STD	Standard Deviation *(p. 33–35, 37, 42, 55)*
TPE	Tree-Structured Parzen Estimator *(p. 19)*
UCB	Upper Confidence Bound *(p. 20–22, 33, 37)*
US	Uncertainty Sampling *(p. 20, 21, 33, 50–52, 56–58)*

1 Introduction

1.1 Motivation

In modern engineering, product development frequently entails the calibration of systems characterised by numerous parameters. Although high-level performance objectives are often well defined, the influence of individual parameters on these objectives is rarely straightforward. This complexity arises from the intricacies of the underlying systems, which in the context of this work are mechatronic in nature. As the number of tunable variables grows, the calibration task becomes increasingly complex, rendering the search for an optimal configuration through exhaustive testing prohibitively time-consuming and costly.

In order to manage this complexity, Design of Experiment (DoE) methods can be employed to systematically select a subset of all possible parameter combinations for evaluation. This enables the construction of mathematical models that capture the relationship between inputs and outputs without necessitating testing every possible combination. These models can then serve as Surrogate Models (SMs) to guide optimisation. Nevertheless, in the context of high-dimensional systems, traditional DoE strategies rapidly become inefficient, as a substantial number of measurements are conducted without yielding new relevant information. Moreover, test environments are often designed to process a predefined experiment plan sequentially without human interaction, making it difficult to adapt the experiment design online. This can lead to a suboptimal use of test resources. However, with increasing digitalisation and the integration of Machine Learning (ML) methods in calibration workflows, the implementation of Adaptive Design of Experiment (aDoE) becomes increasingly feasible.

A highly relevant and practical domain in which these challenges emerge is vehicle calibration, for example in optimising drivability or engine behaviour. In recent decades, there has been a considerable increase in the number of Electronic Control Units (ECUs) and adjustable calibration parameters in new vehicles, which allow the improvement of performance, emissions, comfort, and efficiency [94]. This has resulted in a growing complexity of the fine-tuning process of vehicle dynamics, particularly for premium car manufacturers such as Mercedes-Benz Group AG (MB), where it is paramount to maximise drivability and therefore find the best compromise between dynamics and comfort.

Drivability, in the context of this work, is defined as the perceived quality of a vehicle's longitudinal dynamic response as experienced by the driver, encompassing factors such as smoothness, responsiveness, and overall control [21–23]. Despite its subjective nature, it is a critical product quality that engineers strive to optimise. At present, this optimisation is heavily dependent on empirical, trial-and-error-based manual procedures. Engineers generally alternate between test-rig measurements and test-track validations, making incremental adjustments to parameters to achieve the desired behaviour. This manual and iterative process is characterised by high resource requirements, extended duration, and limited scalability.

Recent research therefore explores the automation of this process using aDoE strategies. Such methods iteratively update a SM and use it to propose new, informative measurement points. In principle, aDoE offers several potential benefits to drivability optimisation: (i) improved data efficiency by reducing the number of samples required to build sufficiently accurate SMs,

(ii) accelerated convergence towards optimal driving behaviour by focusing measurements in promising regions, and (iii) reduced dependence on manual expert intervention. However, it remains unclear which of these theoretical advantages can be realised under real test-rig conditions. This motivates a systematic investigation of the practical impact of aDoE in the drivability calibration workflow.

1.2 Objective and Contribution

While the theoretical potential of aDoE methods has been demonstrated in various research contexts, as elaborated in Section 2.7 presenting relevant work, the practical integration of these methods into existing vehicle drivability optimisation workflows remains largely unexplored. This work aims to bridge this gap by developing an evolutionary, integration-focused approach that transforms the static-DoE methodologies currently used for drivability optimisation into adaptive strategies, while preserving established infrastructure and expertise.

The central objective is to assess how existing DoE methodologies and model architectures can be effectively adapted into an aDoE framework that enhances the efficiency of drivability calibration. Rather than proposing entirely new approaches, this work focuses on evolving current practices to incorporate adaptive elements in a way that remains compatible with the existing test-rig infrastructure and modelling expertise. The evaluation examines improvements in both SM quality and optimisation performance, with particular emphasis on achieving high-quality solutions with fewer measurements while maintaining operational reliability.

To achieve this, several complementary steps are undertaken:

- **Evolution of DoE strategies:** Existing DoE approaches are systematically extended with adaptive capabilities. The performance improvements are evaluated against current static approaches, focusing on both model accuracy and optimisation effectiveness while maintaining compatibility with existing modelling architectures.
- **Integration-focused framework:** A framework is designed to seamlessly integrate adaptive capabilities into the existing test-rig infrastructure. Rather than replacing current systems, it augments them with automated experiment design and model-updating capabilities while preserving established workflows and tools.
- **Real-world validation:** The framework is applied to optimize drivability for the Change-of-Mind Engine Start (COM-ES) manoeuvre and the solutions are checked for plausibility through subjective on-track assessments.

This work therefore builds upon the conceptual advances of earlier aDoE research but introduces a shift in perspective and constraints. The primary goal is to minimise the time between experiment start and completion by enabling continuous, efficient test execution within the existing drivability optimisation workflow. Achieving this goal requires an adaptive framework that can generate new measurement candidates without delaying the measurement process and that supports asynchronous model updates, thereby maximising the efficient use of test-rig time.

This work contributes to the understanding and practical realisation of adaptive design of experiments in the context of automotive drivability calibration. By explicitly evaluating the different mechanisms through which aDoE can provide benefits, it provides a foundation for introducing data-driven, adaptive optimisation workflows into existing development processes.

1.3 Structure of this Work

Chapter 2 introduces the foundational concepts and background necessary to understand the research context. Chapter 3 details the simulation results and their analysis. Chapter 4 describes the experimental setup. Chapter 5 presents the results and evaluates the implications of the proposed approach. Chapter 6 outlines potential avenues for further research. Finally, Chapter 7 summarises the findings and contributions of this work.

2 Theoretical Foundations

In order to answer the research question proposed in Section 1.2, namely whether aDoE can benefit drivability optimisation, a structured experiment must be designed. This in turn requires several theoretical and methodological foundations, which are introduced in this chapter.

This Chapter first defines the context of drivability optimisation in Section 2.1. It then introduces SM foundations in Section 2.2, DoE principles in Section 2.3, and multi-objective Optimisation (MOO) concepts in Section 2.4. Following that the synthetic test function used in the simulativ experiment later on will be defined in Section 2.5. The permutation ANOVA method used in this work is described in Section 2.6. Finally, related literature is reviewed in Section 2.7.

2.1 Drivability Optimisation

2.1.1 Drivability manoeuvres and the status-quo optimisation workflow

This work addresses the optimisation of vehicle drivability, with a particular emphasis on manoeuvres that influence longitudinal dynamics. Drivability manoeuvres can be categorised into five classes: load changes, gear shifts, torque-converter launch, clutch launch, and engine start from coasting [95]. Over recent decades, the number of Electronic Control Units (ECUs) in vehicles has increased substantially, enabling more precise powertrain control and thereby improving performance, efficiency, and drivability [94]. These ECUs comprise a highly complex hierarchy of functions which, among other tasks, coordinate the various drivability manoeuvres, each of which are affected by multiple parameters which can be tuned. This functional and parametric flexibility introduces the challenge of identifying suitable parameter settings that realise the desired driving behaviour. A comprehensive overview of dynamic vehicle manoeuvres is provided in [1].

Drivability optimisation is traditionally performed in later development stages, as it can only be adequately assessed in the real vehicle and cannot be reproduced reliably in virtual simulations [21, 22]. Historically, calibration has relied heavily on manual trial-and-error, whereby application engineers iteratively tune parameters, evaluate them on a test-track or, more recently, on a test-rig, and refine the settings based on their intuition. This workflow is labour-intensive, time-consuming, and entails the risk of exploring the parameter space only locally around established baseline configurations.

Common workflows for drivability optimisation based on DoE experiments are structured as follows. First, a drivability-relevant manoeuvre is specified together with one or more target functions that represent the objectified drivability goals, which often encompass both dynamic and comfort-related criteria as well as emission-related targets. Most manoeuvres can be described by tens of calibration parameters with varying influence, which need to be optimised.

As a first step, an initial experiment is conducted to define both the operation-point space and the design space for the parameters to be optimised, and to identify which parameters have the greatest impact on the target functions. Operation parameters are quantities that are typically controlled by the driver (e.g. vehicle speed, brake pedal position) or determined by external conditions (e.g. road slope, battery voltage, temperature), and thereby define the operating environment of the vehicle. Some of these operation parameters are held constant as they have

no influence on the manoeuvre, but others are varied to ensure that the optimisation results are robust across a range of realistic conditions. The operation-parameter space must be defined carefully so that it accurately restricts the operating conditions to those scenarios that lead to the execution of the selected manoeuvre, while excluding combinations that would result in a different manoeuvre and thus waste experimental resources.

Once the operation-parameter and design spaces are defined, a second DoE experiment is set up with the goal of collecting data to build a model of the drivability targets over the design space. Usually, several iterations are conducted in which the design space is tuned and static-DoE plans are iteratively adapted until models with the desired model quality can be fitted to continue with the optimisation. The resulting SM is typically implemented as a Gaussian Process Regressor (GPR) model. This SM then serves as the basis for MOO, which is commonly carried out using an Non-dominated Sorting Genetic Algorithm II (NSGA-II) algorithm in order to obtain a Pareto Front (PF) of suitable parameterisations. The resulting candidates are, as previously mentioned, validated on the test-track or test-rig and subjectively evaluated.

The final parameterisation is chosen subjectively by the application engineer, based on both the objective results and their subjective assessment of drivability. Model construction and optimisation are carried out using internal tools of MB. The overall workflow is visualised in Figure 2.1.

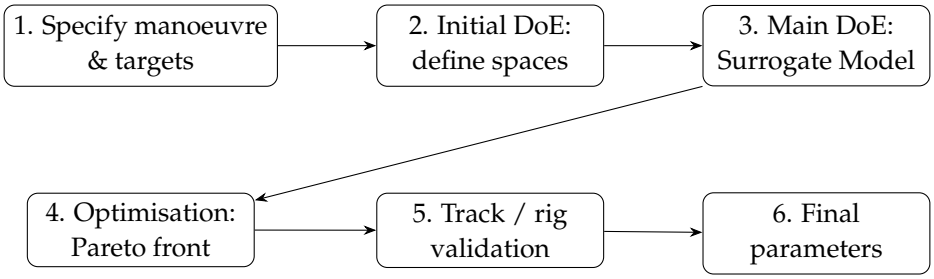


Figure 2.1: Baseline DoE-based workflow for drivability optimisation.

This work focuses on step 3 of the workflow shown in Figure 2.1 as the primary leverage point for methodological improvements. While all steps in the workflow involve substantial human interaction, the construction of the SM in step 3 is currently based on static-DoE designs and manually iterated design-space adjustments and therefore appears to be the most amenable to automation. In its present form, this step lacks the ability to autonomously refine the operation-parameter and design spaces, to steer sampling efficiently towards regions with favourable objective values, and to avoid repeated sampling in regions of consistently poor drivability. These shortcomings motivate the investigation of aDoE-based extensions of step 3 in the following chapters, with the aim of improving data efficiency and optimisation performance within the existing drivability calibration workflow.

2.1.2 Multi-objective nature of drivability optimisation

A major challenge is the inherently multi-objective nature of drivability optimisation. Higher dynamics, such as shorter manoeuvre durations or faster torque build-up, often increase jerk or vibration, which negatively affects perceived comfort. Moreover, as already mentioned above, drivability often depends strongly on the operation parameters as introduced above Hence an optimised calibration must perform robustly across a wide operational domain rather than at a single point. Consequently, the goal is not a single global optimum, but a Pareto-optimal set

representing suitable trade-offs between conflicting objectives. Section 2.4 will cover this more detailed later on.

2.1.3 Model-based optimisation and adaptive Design of Experiment for drivability

Model-based optimisation has increasingly been studied to address these challenges. SM have proven capable of approximating drivability-relevant behaviour and supporting optimisation decisions [95, 24–27]. However, constructing accurate SMs with the aim of finding optima heavily depends on well placed measurements in a high-dimensional design space.

Static-DoE techniques remain common in industrial workflows, but they typically aim at space-filling designs that may include many measurements in behaviourally irrelevant regions at the cost of prediction accuracy near the spots where the optima are. Hence static-DoE often requires multiple human-in-the-loop iterations manually tailoring the experiment design until satisfactory optimisation performance is achieved.

To mitigate these inefficiencies aDoE methods might be beneficial for drivability calibration. By iteratively adapting the experiment design based on the SM predictions and associated uncertainties, aDoE has the potential to concentrate measurements on behaviourally relevant regions and thereby identify high-performing parameter settings with fewer test-rig runs [28]. This could also contribute to bringing down fixed costs for drivability optimisation and could lead to a quicker product development process.

2.1.4 Change-of-Mind Engine Start Manoeuvre

The specific use case studied in this work is the Change-of-Mind Engine Start (COM-ES) manoeuvre. This manoeuvre frequently occurs in urban driving, for example when the vehicle decelerates towards a red traffic light and then re-accelerates before coming to a complete stop as the traffic light turns green [1, 29]. During deceleration, the engine is shut off to reduce fuel consumption and emissions, and it must be restarted and synchronised with the drivetrain once acceleration is requested by the driver again. Poor synchronisation or delayed torque delivery can lead to noticeable jerk and degraded drivability perception.

A schematic overview of a representative COM-ES execution is shown in Figure 2.2. The plot illustrates the characteristic sequence of events that define the manoeuvre and highlights the coupling between driver input, vehicle acceleration, and engine and gearbox speed. These events structure the transient behaviour that optimisation must address, in particular the modelling of the engine speed to result in a smooth acceleration of the car.

Events (1)-(6) in Figure 2.2 correspond to the typical progression observed during a COM-ES manoeuvre which is explained in detail:
The manoeuvre is initiated with the driving car braking with a brake torque of T_{Brk} until the vehicle speed reaches v_{Brk}. In that time the engine is still engaged and therefore dragged by the inertia of the vehicle. At event (1) $v_{\mathrm{Brk}} = 12\,\mathrm{km\,h^{-1}}$ is reached and the brakes get released. The vehicle transitions into a coasting state without driver torque demand, as can be seen in the upper subplot. In the bottom subplot the engine speed curve starts to drop below the transmission input speed curve as the engine decouples and the shut off sequence is initiated. At event (2), the engine shut-off is completed and the engine has stopped turning as indicated by the Engine speed curve dropping to 0 in the bottom subplot, while the vehicle continues to decelerate slightly due to aerodynamic and friction losses, visible in the second subplot where the jerk curve still has a small negative slope. At event (3), the driver applies 40 % throttle at $9\,\mathrm{km\,h^{-1}}$, initiating the COM-ES manoeuvre and activating the drivability function `Phase`

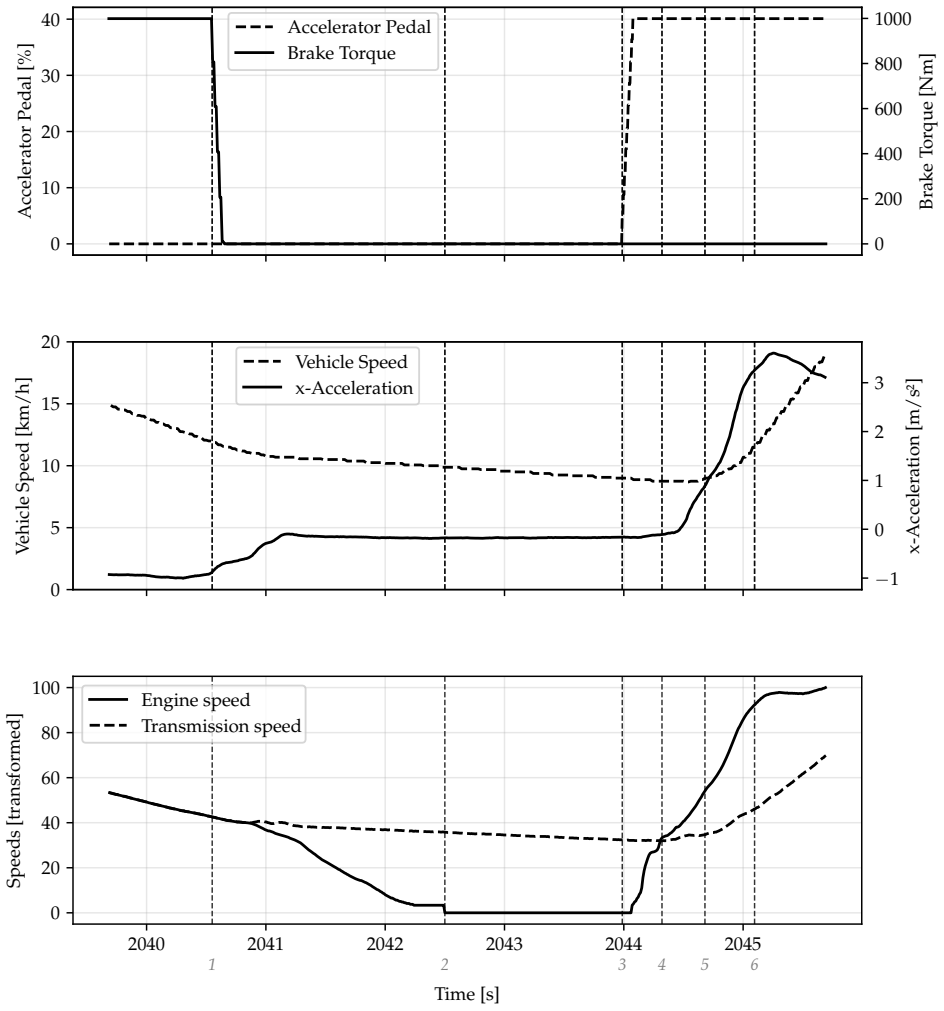


Figure 2.2: COM-ES Manoeuvre Schematic. The numbered vertical lines mark characteristic events during the manoeuvre: (1) brake release and initiating coasting, (2) engine shut-off completed, (3) acceleration driver request, (4) engine restart `Phase 1` completed, (5) engine speed lift `Phase 2` completed, (6) $0.9 \cdot a_{\max}$ is reached.

1. `Phase 1` increases the engine speed to approach the gearbox input speed and prepares the synchronisation process and torque transfer via the hydrodynamic converter [2]. Event (4) marks the termination of `Phase 1` and the onset of `Phase 2`, where the engine speed overtakes the gearbox speed, the torque converter transitions under load, and the vehicle begins to re-accelerate. Event (5) marks the end of the `Phase 2` function. At event (6) $0.9 \cdot a_{\max}$ is reached, which marks the end of the timeframe considered for the evaluation of the COM-ES manoeuvre in this work.

Parameterisation of the COM-ES manoeuvre

The COM-ES manoeuvre is parameterised by eleven variables. Three variable operation parameters control the driving environment during the manoeuvre:

- v_{Brk} ($\mathrm{km\,h^{-1}}$): vehicle speed at brake release,
- v_{App} ($\mathrm{km\,h^{-1}}$): vehicle speed when re-accelerating,
- T_{Brk} (N m): brake torque applied until v_{Brk} is reached.

The remaining eight parameters represent calibration variables governing gearbox-speed gradients, which directly influence the shape of the engine and transmission speed curves and therefore indirectly influence the jerk curve in the middle subplot on which the comfort criterion is based. All variable parameters for the COM-ES manoeuvre are listed in Table 2.1 and are optimised in this work. The table also defines the constraints of the design space within which the experiment is subsequently executed.

Table 2.1: COM-ES Parameter Bounds and associated inequality constraints. (values have been transformed for obfuscation)

Parameter	Type	Lower	Upper	Unit	Constraint
v_{Brk}	Operation Point	3	15	$\text{km}\,\text{h}^{-1}$	$\geq v_{\text{App}}$
v_{App}	Operation Point	3	15	$\text{km}\,\text{h}^{-1}$	$\leq v_{\text{Brk}}$
M_{Brk}	Operation Point	250	2 500	N m	—
$\text{Pha1}_{0\text{–}70}$	Parameter	25	100	$1/\text{s}^2$	$> \text{Pha1}_{90\text{–}95}$
$\text{Pha1}_{90\text{–}95}$	Parameter	3.75	50	$1/\text{s}^2$	$< \text{Pha1}_{0\text{–}70}$
Pha2_0	Parameter	3.75	100	$1/\text{s}^2$	$< \text{Pha2}_{10}$
Pha2_{10}	Parameter	3.75	100	$1/\text{s}^2$	between Pha2_0 and Pha2_{30}
Pha2_{30}	Parameter	3.75	100	$1/\text{s}^2$	between Pha2_{10} and Pha2_{40}
Pha2_{40}	Parameter	10	100	$1/\text{s}^2$	between Pha2_{30} and Pha2_{50}
Pha2_{50}	Parameter	10	100	$1/\text{s}^2$	between Pha2_{40} and Pha2_{95}
Pha2_{95}	Parameter	10	100	$1/\text{s}^2$	$> P_{50}$

Test-rig execution and optimisation objectives

The manoeuvre is automatically executed on the test-rig by first accelerating the vehicle to a constant $50\,\text{km}\,\text{h}^{-1}$. This is followed by a deceleration phase in which a brake torque of T_{Brk} is applied until the vehicle speed has reduced to v_{Brk}, after which the vehicle coasts until it has further decelerated to v_{App}, before acceleration is again requested by the driver through a 40 % throttle application. This throttle application is fixed for this thesis, because the `Phase 1 & 2` functions use different parameter values for different throttle application levels. In this work, the COM-ES manoeuvre with a 40 % throttle application is used as an illustrative example for optimisation. Other throttle application levels could be optimised in a similar manner but might leed to different optimal parameter settings.

The restart behaviour during this interval is critical for perceived drivability, as illustrated by the transitions in Figure 2.2. The dynamics objective is defined as the time elapsed between event (3) and event (5). The comfort objective is derived from the longitudinal-acceleration, also called jerk, time series and quantifies its smoothness in the interval starting from event (3) until $0.9\,a_{\text{max}}$ is reached, in this case at event (6), following the ISO 2631-1:1997 standard [100]. These objectives are inherently conflicting and necessitate optimisation towards a Pareto-optimal compromise.

2.2 Surrogate Models

In general, any parametric or non-parametric regression model that can be fitted to the respective experimental data can serve as a SM in the context of aDoE. A SM is usually a computationally cheap approximation of a prohibitively expensive input-output relation, constructed from a limited set of evaluations of the underlying system [3, 4]. This requirement becomes partic-

ularly stringent in experiments where individual measurements are comparatively quick, as it would defeat the purpose of aDoE if a significant fraction of the total experimental budget were consumed before the SM is updated even once. Ideally, the computational effort required to fit or update the SM should be comparable to the time needed for only a few experiments, so that the model can be refreshed frequently and promptly provide new candidate factoring in the knowledge gained from the past measurements.

Formally, any mathematical model representing a input output relationship can be writte as follows: With the input parameters as $\mathbf{x} \in \mathbb{R}^d$, Y as the unknown response of the system and ε as the noise term, the system can be described by:

$$Y = f(\mathbf{x}) + \varepsilon, \tag{2.1}$$

where $f(\mathbf{x})$ is the unknown deterministic component of the response and ε is a random noise term with $\mathbb{E}[\varepsilon] = 0$ and $\mathrm{Var}(\varepsilon) = \sigma^2$. Since $f(\mathbf{x})$ is rarely available in closed form or may be prohibitively expensive to evaluate, we approximate it by a fitted SM $\hat{f}(\mathbf{x})$ and write

$$f(\mathbf{x}) = \hat{f}(\mathbf{x}) + \varepsilon, \tag{2.2}$$

where, in this context, ε collects model inadequacy, unmodelled physics, and measurement error in addition to intrinsic process variability [5, p. 8].

2.2.1 Model Architectures

While generally any Machine Learning (ML) model can be used as a SM, certain architectures are more suitable for this task than others, depending on factors such as model fitting duration, inference time, and the ability to estimate prediction uncertainty. Commonly used model architectures for aDoE include:

- **Polynomial Models:** Suitable only for simple, mostly linear or low-dimensional systems. Their limited flexibility makes them inadequate for capturing complex nonlinear relationships [30].
- **Random Forest (RF):** Efficient and easy to fit with limited computational resources. Random Forests perform robustly on medium-sized datasets and require little hyperparameter tuning. However, their performance degrades when input features are weakly correlated with the target variable, and they provide only approximate measures of predictive uncertainty derived from the prediction variance of the individual trees [31].
- **Gaussian Process Regressor (GPR):** Highly flexible for regression tasks and well-suited for uncertainty modeling, making it a common SM in aDoE tasks. However, training scales cubically with the number of data points ($\mathcal{O}(n^3)$), which makes it very quick to fit for small datasets but limits its practicality for large datasets [32, 96, 97].
- **Local Model Network (LMN):** Nonlinear model structures used in certain industrial and process-control applications. While they can capture system dynamics effectively, their high complexity and parameter sensitivity make them less suitable for aDoE optimisation tasks where computation has to be quick [33].

Gaussian Process Regressor

A GPR, also known as a Kriging model [34–36, 6], is a non-parametric Bayesian approach to regression that defines a probability distribution over functions. Kriging was originally developed in geostatistics and can be interpreted as GPR with particular choices of covariance

structure. Instead of learning a finite set of weights as in linear or neural network models, a Gaussian Process represents functions implicitly through a covariance function (kernel), which expresses how outputs at different inputs are correlated. Intuitively, if two inputs are close under the kernel, their outputs are expected to be similar, whereas inputs that are far apart under the kernel are treated as less correlated. This makes GPR highly flexible and yields uncertainty estimates that are internally consistent with the underlying probabilistic model as an integral part of the prediction. In the following, the mathematical formulation of GPR is introduced following [5] and the documentation of the python package scikit-learn[1], which is the implementation used in this work.

A Gaussian Process is fully determined by a mean function $m(x)$ and a covariance function $k(x, x')$,

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

which implies that for any finite set of inputs $X = [x_1, \ldots, x_n]$ the function values follow a multivariate normal distribution,

$$f(X) \sim \mathcal{N}(m(X), K(X, X)), \qquad K_{ij} = k(x_i, x_j).$$

In practice, the mean function is usually taken as $m(x) = 0$ for the prior assumption, while the kernel encodes assumptions such as smoothness or characteristic lengthscales of variation. The lengthscale determines how quickly correlations decay in the input space: a small lengthscale allows rapid variation of the function whereas a large one leads to smoother behaviour. Separate lengthscales for each input dimension, often referred to as an anisotropic kernel, allow the model to learn which input dimensions are more relevant, which is also known as Automatic Relevance Detection (ARD) in many ML frameworks. In this setting, ARD corresponds to learning one lengthscale per input dimension, where large lengthscales effectively down-weight features that are less relevant for explaining the response.

In regression, we observe a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ with input locations x_i and corresponding scalar outputs (labels) y_i. Collecting the inputs in the matrix $X = [x_1, \ldots, x_n]$ and the outputs in the vector $y = [y_1, \ldots, y_n]^\top$, the observations are modelled as

$$y = f(X) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma_n^2 I),$$

so that, under this model, the noisy outputs y follow a Gaussian distribution with covariance $K(X, X) + \sigma_n^2 I$, where σ_n^2 denotes the noise variance. For notational convenience, define

$$K_y(X, X) = K_\theta(X, X) + \sigma_n^2 I,$$

where $K_\theta(X, X)$ is the kernel matrix evaluated with kernel hyperparameters θ (e.g. lengthscales and signal variance), and the observation-noise variance σ_n^2 is treated separately. The marginal likelihood of the training labels y under the GPR is then

$$\log p(y \mid X, \theta, \sigma_n^2) = -\frac{1}{2} y^\top K_y^{-1} y - \frac{1}{2} \log |K_y| - \frac{n}{2} \log 2\pi,$$

where $K_y = K_y(X, X)$ for brevity. Maximising this quantity trades off data fit (first term) against model complexity (second term): the log-determinant term penalises models that assign very

[1] `https://scikit-learn.org/stable/modules/gaussian_process.html#gaussian-process-regression-gpr` (visited 15.11.2025)

low prior probability mass to the observed data. In scikit-learn, these hyperparameters are treated as deterministic and are chosen by maximising the log marginal likelihood using the L-BFGS-B algorithm, which will be revisited later in Section 2.4. As a result, the predictive variance is computed using point estimates of the hyperparameters [37], corresponding to an empirical-Bayes approximation in which uncertainty in (θ, σ_n^2) is not propagated into predictions. Computationally, the dominant cost is the $\mathcal{O}(n^3)$ Cholesky factorisation of the covariance matrix K_y, required both for evaluating the log marginal likelihood and for making predictions; memory usage scales as $\mathcal{O}(n^2)$ due to storage of the covariance matrix. How Cholesky factorisation works can be read in detail in [38].

Once the hyperparameters are optimised, prediction at a new test input x_* is obtained by conditioning the joint Gaussian distribution of the training outputs y and the latent function value $f(x_*)$ on the observed data. Equivalently, this corresponds to computing the posterior distribution $p(f(x_*) \mid X, y, \theta, \sigma_n^2)$, where X and y are the training inputs and their associated labels. This yields a Gaussian predictive distribution for the latent function,

$$f(x_*) \mid X, y \sim \mathcal{N}(\mu_*, \sigma_*^2),$$

with mean and variance

$$\mu_* = k_*^\top K_y^{-1} y, \qquad \sigma_*^2 = k(x_*, x_*) - k_*^\top K_y^{-1} k_*,$$

where $K_y = K_\theta(X, X) + \sigma_n^2 I$ is the covariance matrix of the noisy observations and $k_* = k(X, x_*)$ is the vector of covariances between the training inputs X and the test input x_*. In the mean expression, the quantity $K_y^{-1} y$ can be interpreted as a set of weights assigned to the training labels, and $k_*^\top$ combines these weights according to the similarity between x_* and each training point as encoded by the kernel. Thus, the mean prediction is a similarity-weighted combination of the observed outputs, where similarity is defined by the kernel. The predictive variance σ_*^2 decomposes into the prior variance at x_*, given by $k(x_*, x_*)$, minus the reduction in uncertainty due to the information contained in the training data through the term $k_*^\top K_y^{-1} k_*$. As a result, the predictive variance expresses the uncertainty of the SM, which is small near training points and grows in regions where the data provide little information. If interest lies in the predictive distribution of a future noisy observation $y_* = f(x_*) + \epsilon_*$ rather than the latent function value, an additional noise term is added and the predictive variance becomes $\mathrm{Var}(y_*) = \sigma_*^2 + \sigma_n^2$.

Different kernels encode different structural assumptions. Common examples include the Radial Basis Function (RBF) kernel, which produces very smooth functions, the Matérn kernel, which allows adjustable smoothness via its parameter ν, and the Rational Quadratic (RQ) kernel, which can model variability at multiple characteristic lengthscales [98]. Kernels may be added or multiplied to build more expressive models, such as combining periodic components with long-term trends. Examples are shown in Table 2.2. When noise is modelled explicitly by including a WhiteNoise component, an additional diagonal term $\sigma_w^2 I$ is added to the covariance matrix. In the notation above, this corresponds to choosing $\sigma_w^2 = \sigma_n^2$ so that $K_y(X, X) = K_\theta(X, X) + \sigma_w^2 I$. The predictive mean remains governed by the smooth kernel, while the predictive variance σ_*^2 is increased by a noise contribution at each input location. As a consequence, σ_*^2 reflects both uncertainty about the latent function and the assumed observation noise, and the model is no longer forced to interpolate the training data exactly, but instead yields smoother mean predictions with confidence intervals that do not collapse to zero at the observed points.

The implementation in scikit-learn performs exact gaussian process inference. This leads to $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ memory complexity, limiting exact GPRs to datasets of moderate size. Within this regime, the approach retains its full probabilistic interpretation and provides accurate predictions together with meaningful uncertainty estimates if the kernel and noise hyperparameters are well calibrated to the data.

Figure 2.3a shows an example of a GPR fitted to observations of a one-dimensional function and the 95 % confidence interval based on the predicted uncertainty around the mean prediction. For comparison, an Random Forest (RF) ensemble model is shown in Figure 2.3b, fitted to the same data with the 95 % confidence interval estimated from the standard deviation of the individual tree predictions. It can be seen that the predictions of the GPR, as well as the associated uncertainty, form continuous functions, whereas the RF exhibits discontinuous jumps in its uncertainty. These jumps can cause multiple candidates to be assigned the same uncertainty, which may be undesirable when ranking candidates based on uncertainty or when using gradient-descent-based Acquisition Function (AF) optimisation in applications such as bayesian optimisation or active learning.

2.2.2 Model Evaluation Metrics

Quantifying model quality is essential for assessing its utility and ensuring that it reliably supports improvements in system behaviour. To enable such assessment, test criteria are required to evaluate and compare different modelling approaches. For this purpose, the Coefficient of Determination (R^2) shown in Equation 2.3 and the Root Mean Square Error (RMSE) defined in Equation 2.4 are commonly used [101].

For proper evaluation, it is necessary to hold back a portion of the available data and use it as a validation or test dataset. This allows the model to be evaluated on unseen data is a good indicator if the model is actually learning something from the data and if it is overfitting. The model is therefore primarily assessed based on its performance on this unseen data. In case of a GPR no validation data is needed for the Hyper Parameter Optimisation (HPO), so the validation data can be fully used for model assessment.

The R^2 metric is a common metric which quantifies the proportion of variance in the observed data that is explained by the model. As it ranges from 0 to 1, higher values indicate better model performance, with a value of 1 indicating a perfect fit to the data. This also makes the metric easy to interpret.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2} \tag{2.3}$$

RMSE is another commonly used metric to quantify the difference between predicted values $\hat{y}_i$ and observed values y_i. It provides a measure of the average magnitude of the prediction errors, with larger errors being penalised more heavily due to the squaring of the differences. Lower RMSE values indicate better model performance.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}} \tag{2.4}$$

In this work, RMSE is evaluated using (i) a global validation dataset spanning the complete input space and (ii) a local validation dataset formed by selecting a fixed percentage of the global validation samples that lie closest to the true optimum. This setup allows assessment of both global model quality and local model quality in the vicinity of the optimum.

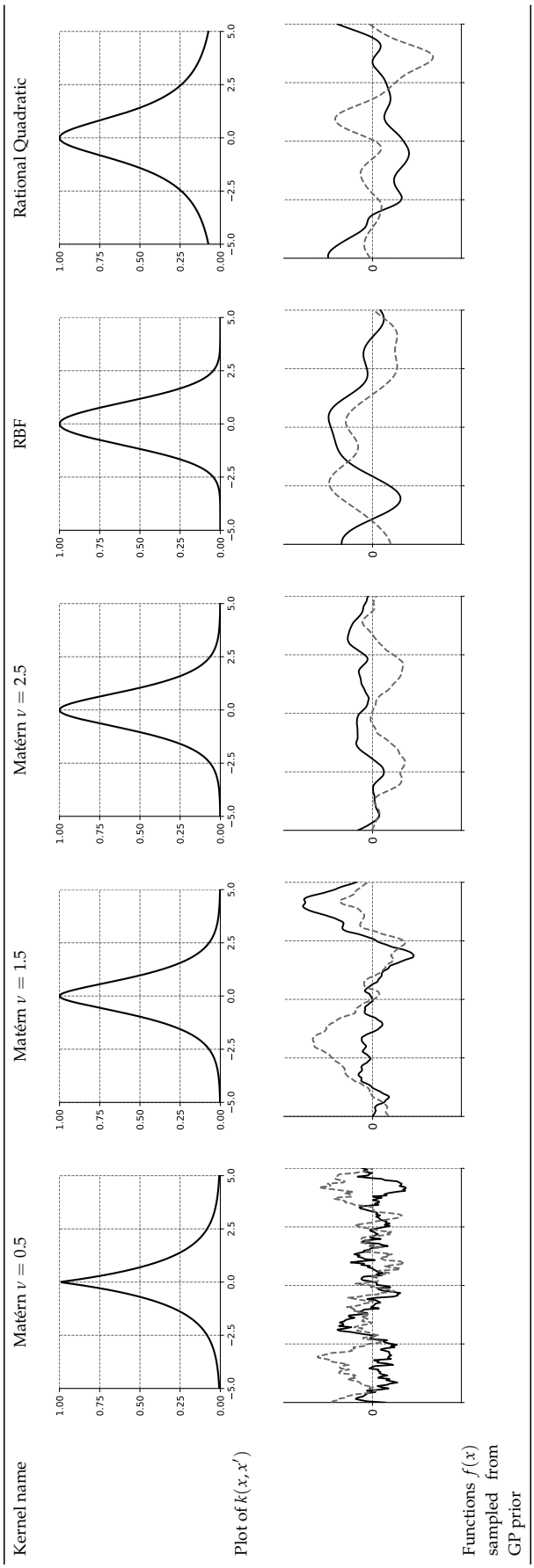
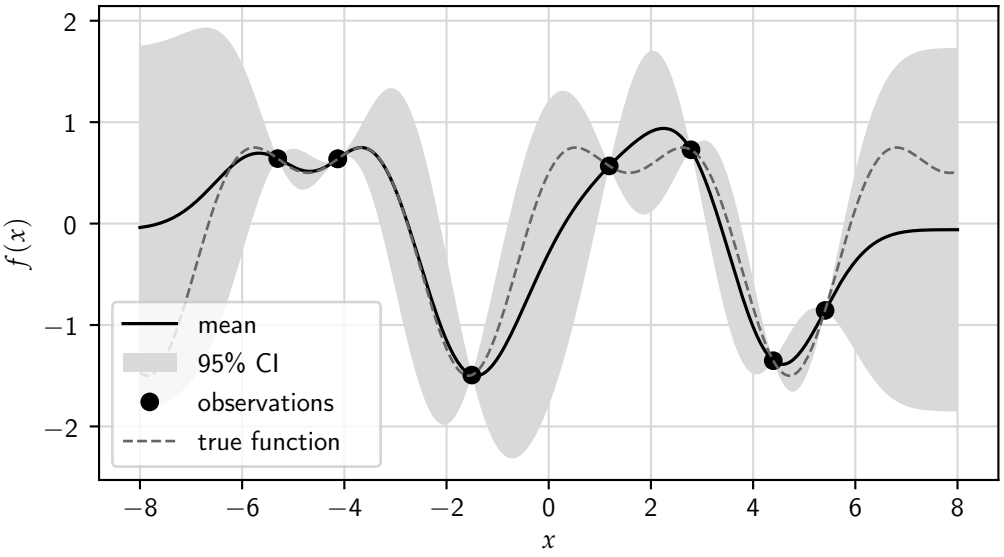
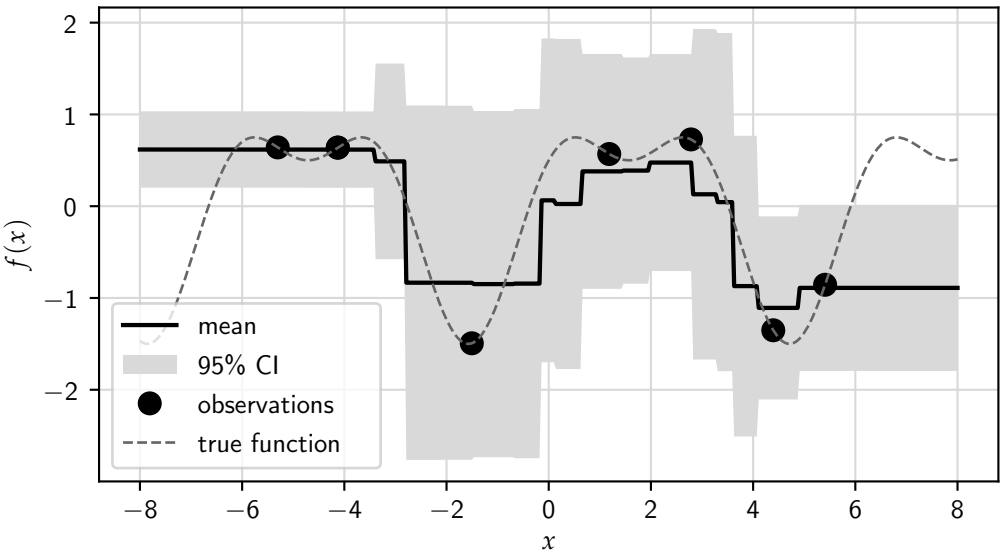


Table 2.2: Examples of structures of basic Gaussian process kernels. (adapted from [98])



(a) GPR example with uncertainty



(b) RF model example with uncertainty

Figure 2.3: Comparison of SMs with uncertainty quantification (adapted from [5, p. 3]).

Additionally, we define the *Gap to Optimum*, which is used to quantify the distance between a predicted optimal target value and the true optimal target value:

$$\text{Gap to Optimum} = |\hat{y}_{\text{opt}} - y_{\text{opt}}| \tag{2.5}$$

where $\hat{y}_{\text{opt}}$ is the predicted optimal target value and y_{opt} is the true optimal target value of the system. Naturally, this metric can only be applied to synthetic benchmarks where the true optimum is known.

2.3 Design of Experiment

Design of Experiment (DoE) concerns planning how many measurements are required to obtain a model that approximates the system response with sufficient accuracy. Accuracy is typically assessed using metrics such as R^2 or RMSE as introduced in Section 2.2.2. The required number of samples strongly depends on both the type of model used to approximate the system response and the complexity of the underlying system. A system with an approximately linear response surface and weak interaction effects between parameters will typically require fewer samples than a highly nonlinear system with many peaks and valleys.

In the simplest case with k discrete parameters, where parameter j can take on k_j distinct levels, exhaustively sampling every possible combination of parameters using a Full Factorial Designs (FFD) provides complete coverage of the design space and perfect knowledge of the system response, assuming no measurement noise, and corresponds to the *No model* case in Table 2.3. If the size of the input space is prohibitively large and measurements are time-consuming, it is necessary to reduce the number of samples and devise an efficient sampling strategy [7]. The exponential increase in sample size with the number of parameters is often referred to as the curse of dimensionality [8, p. 190]. If, for example, a parametric model such as a polynomial regression model is to be fitted to the sampled data, the required number of samples must at least match the number of model coefficients, which grows combinatorially with the polynomial degree and the number of features, as summarised in the *Polynomial of degree n* row of Table 2.3. For more complex, non-parametric models, a frequently cited rule of thumb is that on the order of ten samples per parameter are needed in order to obtain a good approximation of the system response, see the *Non-parametric model* row in Table 2.3 [9]. The above rules of thumb, summarised in Table 2.3, provide only rough estimates, and the actual number of samples needed can vary greatly depending on the complexity of the system, the measurement noise, and the model applied to the data, as mentioned above.

Table 2.3: Approximate number of samples required for different model types with input dimension k.

Model Type	Number of Samples
No Model	$\prod_{i=1}^{k} k_i$
Polynomial of degree n	$\binom{k+n}{n}$
Non-parametric model	$10k$

Besides estimating the number of samples needed to fit a meaningful model, another question is how to efficiently choose the samples to be measured. As often no prior knowledge is

available, the area of interest is initially assumed to be the entire design space.

DoE strategies can, at a high level, be divided into two categories:

- *static-DoEs* define all experimental runs in advance and distribute them across the input space. They can be either space-filling or model-based.
- *aDoEs* iteratively update a SM and adapt or extend the DoE according to a strategy, for example by optimising model quality or target variables, or by following a heuristic such as placing new samples close to previous measurements that yielded good results.

In the case of a static-DoE, a table of candidate experiments is defined before running any measurements, and during experimentation all of these experiments are executed. Consequently, none of the information gained during the experiment is used to adapt the following measurements. However, static designs have the advantage of being easy to automate and implement, as the interaction between the test plan and the execution environment is unidirectional, with the environment loading a test plan in the form of a table and executing it sequentially.

An aDoE assumes that, as samples are measured, past measurements can be used to guide subsequent sampling towards regions where it contributes most to optimising model quality or the system's target variables early on [39, 102]. This makes data analysis during the experiment necessary and therefore more complex to implement, automate and reproduce, but it allows for more efficient sampling, as areas of interest can be identified and potentially fewer data points are allocated to uninformative regions.

In this Section, Subsection 2.3.1 reviews static-DoEs, and Subsection 2.3.2 introduces aDoE strategies and their advantages over static-DoE.

2.3.1 Static Design of Experiment

Static-DoE is used in this work as an umbrella term for designs that fix all experimental runs before any data are collected, relying on a predetermined plan to explore the input space. This approach requires no feedback in the form of intermediate results from the testing environment but may collect measurements in regions where the response surface is trivial or uninteresting, for example because they are far away from the optimum. Generally, static-DoEs focus on exploration, as no tuning towards optima is possible due to the lack of any knowledge about the system prior to measuring.

Static-DoEs can be broadly divided into two categories: those that assume no prior knowledge about the system, and those that assume a predefined model, e.g. a linear model, and therefore allow assumptions about system behaviour.

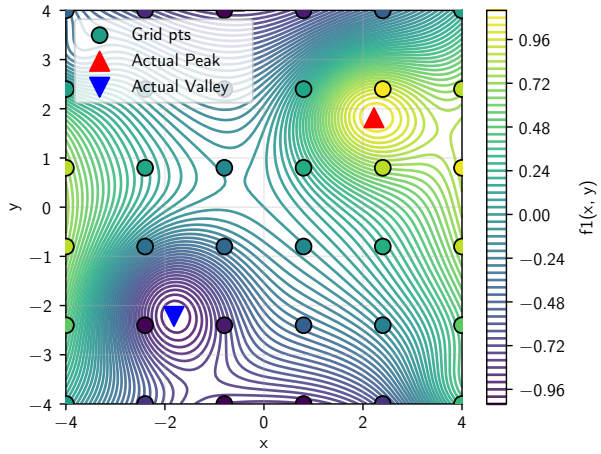
- Examples of methods not assuming any prior knowledge of the model:
  - **Full Factorial Designs (FFD)** tests every combination of parameters, resulting in the same exponential growth in the number of experiments as in the *No model* case in Table 2.3, with k being the number of parameters and k_i the number of discrete states of parameter i. This maximises information but scales exponentially with dimensionality, making it feasible only for small, discretised parameter spaces. [40][10, Ch. 7][8, p. 190]
  - **Grid Sampling** samples points on a predefined grid across the input space, as shown in Figure 2.4a. This ensures coverage of the entire region but becomes inefficient for

complex response surfaces and large design spaces and may systematically miss optima lying between grid points. [41]

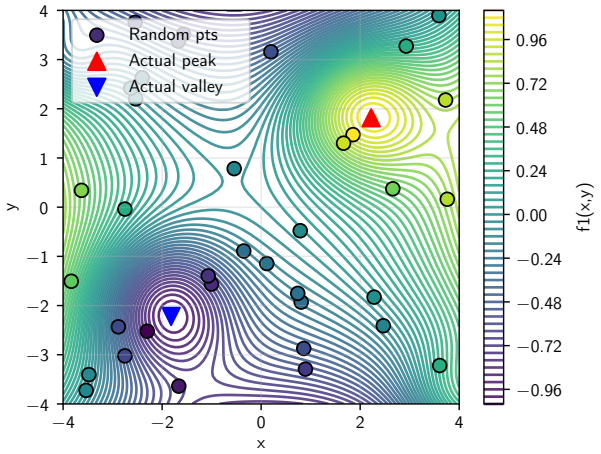
- **Random Sampling** selects experimental runs uniformly at random across the input space, as illustrated in Figure 2.4b. This strategy is easy to implement, does not require prior system knowledge, and provides favourable statistical properties that help avoid systematic bias. However, random sampling may produce clusters or leave gaps, resulting in non-space-filling designs. It is often used when sampling cost is low or when quantifying uncertainty. [42, 43]
- **Latin Hypercube Sampling (LHS)** is a stratified sampling method designed to generate representative and largely uncorrelated samples from a multidimensional continuous distribution. An example is shown in Figure 2.4c. For a given number of samples N, each parameter range is divided into N equally sized intervals, and exactly one sample is drawn from each interval. By enforcing this stratification along every dimension, LHS achieves better coverage of the design space than simple random sampling while preserving randomness within each stratum. Beyond purely random selection inside each interval, optimised LHS variants aim to further improve space-filling properties. A common approach is the *maximin* criterion, in which the sample configuration is chosen to maximise the minimum pairwise distance between points, thereby reducing clustering and distributing samples more uniformly across the domain. [44–47] [11, pp. 205–209]

- Examples of methods where the model to be fitted on the data is already assumed (e.g., polynomial regression), allowing assumptions about the response:
  - **A-Optimality** minimises the trace of the inverse Fisher Information Matrix, which corresponds to minimising the average variance of the estimated parameters. This is useful when the primary goal is precise parameter estimation rather than global prediction quality. [10]
  - **D-Optimality** maximises the determinant of the Fisher Information Matrix, minimising the volume of the confidence region of the estimated parameters and thus reducing overall parameter uncertainty for parametric models (linear or nonlinear regression). [7, p. 442]
  - **G-Optimality** minimises the maximum prediction variance over the design space, ensuring that the worst-case prediction error is as small as possible. In classical optimal design for linear regression models, G-optimal designs are equivalent to D-optimal designs through the Kiefer–Wolfowitz equivalence theorem. [48, 49]
  - **I-Optimality** minimises the average prediction variance over the design space, improving overall predictive performance of the fitted model rather than parameter accuracy alone. [10]
  - **S-Optimality** maximises the average (or integrated) log-determinant of the Fisher Information Matrix across a set of candidate models, providing designs that are robust against model misspecification and improve model discrimination capability. [50]

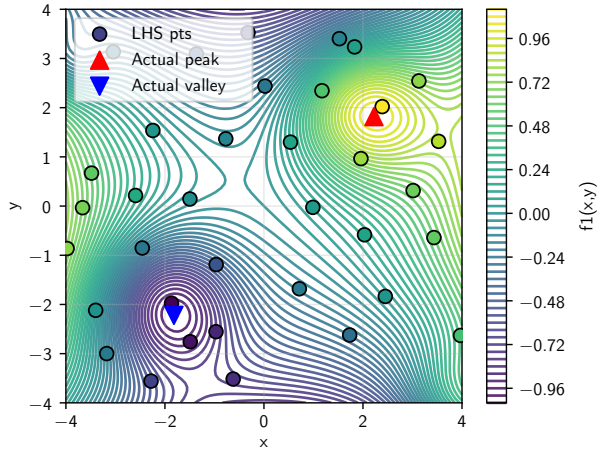
Comparing the different model free strategies in Figure 2.4 reveals some of the differences in their space-filling properties. Grid sampling (Figure 2.4a) provides uniform coverage of



(a) Grid with 36 samples



(b) Random with 36 samples



(c) LHS with 36 samples

Figure 2.4: Comparison of different static sampling strategies: (a) Grid sampling, (b) Random sampling, (c) Latin Hypercube Sampling.

the design space but may systematically miss characteristics of the response surface that lie between grid points. Random sampling (Figure 2.4b) introduces variability and is better suited for capturing complex response surfaces, but it can lead to clustering and gaps in coverage. Latin Hypercube Sampling (LHS) (Figure 2.4c), when compared to random sampling, shows better space-filling properties with fewer clusters.

More detailed information about static-DoE can be found in [7, 10, 11, 51, 12–14, 52].

2.3.2 Adaptive Design of Experiment (aDoE)

In contrast to static-DoE approaches discussed above in Subsection 2.3.1, aDoE continuously updates and generates the DoE on demand without any fixed prior design, based on information from past measurements. This information is typically encoded in a SM, which is used to evaluate a pool of candidates which have not yet been measured by applying an Acquisition Function (AF) which calculates a scalar which can be interpreted as utility. Different types of AFs and SMs will be discussed below. By evaluating candidates using the SM, without ac-

tually executing the measurements, aDoE can focus sampling on informative regions of the design space, thereby reducing unnecessary experiments and saving time and resources. This approach relies on the assumption that the SM captures the true system behaviour sufficiently well to guide the sampling process towards candidates that improve model quality or system performance. Under this assumption, aDoE can achieve comparable or superior performance to static-DoE while requiring fewer measurements.

Nomenclature and Taxonomy of aDoE

Similar principles are discussed across various research communities under different terminologies, including Sequential model-based Global Optimisation [15, 53, 54], Active Learning [55, 103, 16, 56–60], online Design of Experiment [26], and Adaptive Design of Experiment (aDoE) [25]. For consistency, this work uses the term Adaptive Design of Experiment (aDoE) to refer to all methods that iteratively adapt the DoE during the experiment.

Below, some common terms used in the context of aDoE are defined.

- **Acquisition Function:** A mathematical function that assigns a utility value to candidate points based on the expected benefit of sampling them. AFs can favour exploration, exploitation, or a combination of both. Some examples are discussed in more detail in Subsection 2.3.2.
- **Candidate:** A combination of parameters (a point in the design space) that has not yet been measured in a previous trial. During the selection process usually a pool of candidates is sampled in a space filling manner and then evaluated using a AF.
- **Exploitation:** The process of selecting candidates that refine the model around currently promising regions with favourable predicted objective values.
- **Exploration:** The process of selecting candidates in unexplored or highly uncertain regions of the design space in order to improve the overall model quality.
- **Surrogate Model:** A mathematical model fitted to the available measurement data that is used to guide the sampling of new candidates. Examples of SM architectures are discussed in Section 2.2.
- **Trial:** A single execution of the experiment using the parameters of the candidate selected from the candidate pool based on the AF.

One way to organise aDoE approaches is by distinguishing whether they rely on an explicit SM of the objective or operate directly on the observed data distribution.

Model-based methods construct a SM of the objective function from the available measurements and use it to identify new candidates with high AF values. This can be achieved either by sampling a large number of random candidates and evaluating them using the SM, or by directly optimising the AF with an optimisation algorithm. Model-based methods can be further divided into Bayesian and non-Bayesian approaches. Bayesian model-based methods, such as GPR-based Bayesian optimisation or Tree-Structured Parzen Estimator (TPE), explicitly maintain or approximate a posterior distribution over the unknown objective, which is exploited to balance exploration and exploitation when selecting new candidates. Non-Bayesian model-based methods, in contrast, employ parametric regressors or ensembles such as polynomial models, random forests, or support vector machines. Adaptive sampling is then driven by heuristics such as prediction uncertainty, diversity, or residual variance, without explicit pos-

terior inference. For a comprehensive overview of model-based adaptive sampling strategies, and Bayesian optimisation in particular, see [61–63, 104].

Model-free methods do not construct an explicit surrogate of the objective but instead exploit the empirical distribution of the observed data or design criteria directly. This includes resampling strategies that favour regions where good observations have been found, as well as sequential augmentation of classical space-filling designs. Such strategies emphasise exploration or exploitation through geometric or data-driven rules rather than through explicit surrogate or posterior modelling.

Acquisition Functions

AFs define a criterion for ranking candidate points and thus determine where the next measurements are taken. Depending on their formulation, they can be predominantly explorative, predominantly exploitative, or a mixture of both.

Uncertainty Sampling: Uncertainty Sampling (US) is a purely explorative AF, where the next sample is chosen to be the candidate with the highest predictive uncertainty $\sigma(x)$ of the SM. For a model ensemble, this uncertainty can be estimated as the empirical standard deviation of the ensemble predictions, as given in Eq. (2.6). For GPR, $\sigma(x)$ is directly obtained from the predictive variance of the gaussian process.

$$\text{Uncertainty Sampling} = \sigma(x) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(f_i(x) - \mu(x)\big)^2} \tag{2.6}$$

where $f_i(x)$ is the prediction of the i-th model in the ensemble, $\mu(x)$ is the mean prediction of the ensemble, and n is the number of models in the ensemble. An example of US is shown in Figure 2.6a.

Expected Improvement: Expected Improvement (EI) is a strategy which trades off exploration and exploitation, as illustrated in Figure 2.5a. The effect of EI applied to a simple one-dimensional function can be seen in Figure 2.6b. The acquisition value is computed based on the predicted improvement over the current best sample and the associated uncertainty of the SM at that point. The formulation of EI, as proposed by Jones et al. [64], is given by:

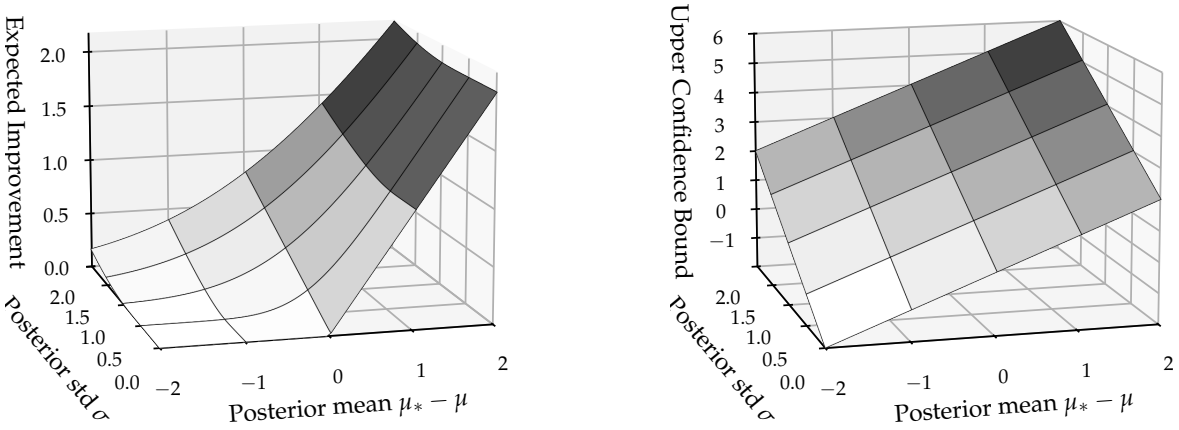
$$\text{EI}(x) = (f_{\text{best}} - \mu(x)) \cdot \Phi\left(\frac{f_{\text{best}} - \mu(x)}{\sigma(x)}\right) + \sigma(x) \cdot \phi\left(\frac{f_{\text{best}} - \mu(x)}{\sigma(x)}\right) \tag{2.7}$$

Here, $\mu(x)$ and $\sigma(x)$ denote the predicted mean and standard deviation of the SM at point x, respectively, and f_{best} represents the best, in this case minimum, objective value observed so far. $\Phi(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution, while $\phi(\cdot)$ denotes its probability density function (PDF). The first term in Eq. (2.7) represents the expected improvement due to exploitation, whereas the second term reflects the contribution of exploration arising from predictive uncertainty.

Upper Confidence Bound: Upper Confidence Bound (UCB) is another AF that balances exploration and exploitation by adding a multiple of the standard deviation to the mean prediction of the SM, as illustrated in Figure 2.5b. The UCB is defined as [65, 105]:

$$\text{Upper Confidence Bound} = \mu(x) + \kappa\sigma(x) \tag{2.8}$$

where $\mu(x)$ is the predictive mean of the SM, $\sigma(x)$ is the predictive standard deviation, and $\kappa > 0$ is a tunable parameter that controls the trade-off between exploration (larger κ) and exploitation (smaller κ). An example of UCB is shown in Figure 2.6c.



(a) EI surface

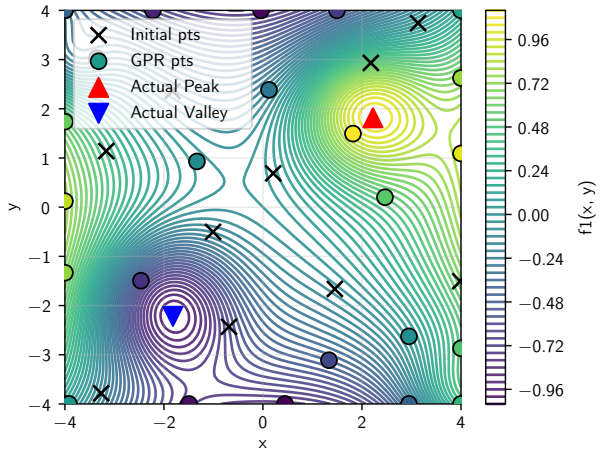
(b) UCB surface with $\kappa = 2$

Figure 2.5: Comparison of acquisition function surfaces. σ is the predicted standard deviation and $\mu_* - \mu$ is the difference between the predicted mean and the best observed value, with larger values indicating greater potential improvement.

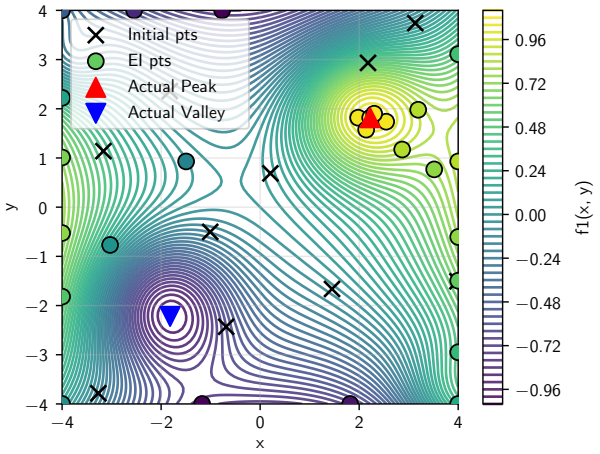
Comparing the different strategies in Figure 2.6, all three strategies can be clearly distinguished. US (Figure 2.6a) shows the highest spread across the design space, which indicates its explorative nature. EI (Figure 2.6b) focuses more on areas around the optimum while still sampling some areas further away, indicating a balance between exploration and exploitation. UCB (Figure 2.6c) shows behaviour similar to EI, but with the chosen κ value of 2 it seems more exploitative, focusing more on areas towards the optimum. Generally, all these methods exhibit a bias towards the edge regions of the input space, which is a known effect and controlled by the estimated uncertainty of the GPR. Still, the exploitative strategies (EI and UCB) both find the optimum in this case and are not misled by this effect. In comparison with Figure 2.4, all three adaptive strategies seem less space-filling than the static strategies, even the explorative uncertainty approach, due to the bias towards the edges of the design space.

Besides the purely exploitative strategy of directly optimising the SM and sampling its predicted optimum, most AFs are primarily designed for single-objective problems. In a multi-objective setting, different targets can exhibit different noise levels ϵ, value ranges, and model errors, which can introduce sampling bias towards specific objectives if the acquisition values are aggregated naively. To extend single-objective AFs to multi-objective problems, which is relevant in the context of this work, different aggregation strategies can be applied:

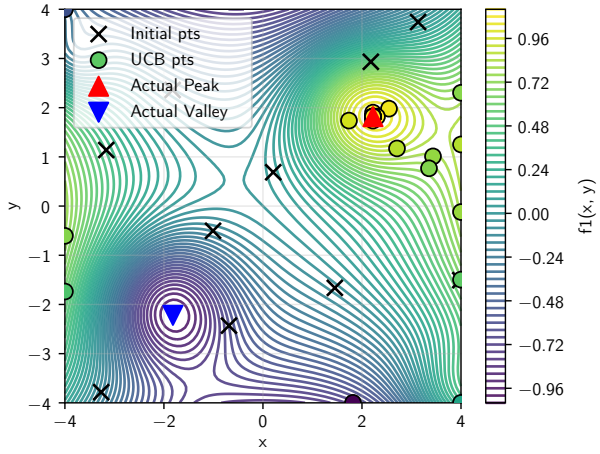
- **Sum:** The acquisition values of all targets are added. If the targets differ in scale, range, or noise level, those with larger scales, broader ranges, or higher noise can dominate the aggregated value.
- **R^2-weighted sum:** The acquisition values of all targets are combined into a single value weighted by $(1 - R^2)$ of each target. This prioritises targets with a low R^2 score, which typically require more samples to improve the model quality.



(a) Uncertainty with 36 samples



(b) EI with 36 samples



(c) UCB with 36 samples

Figure 2.6: Comparison of different adaptive sampling strategies: (a) Uncertainty, (b) Expected Improvement, (c) Upper Confidence Bound.

- **Round robin:** The objectives are sampled in a Round-Robin fashion. The next sample is chosen as the candidate with the highest acquisition value for the objective that has not been sampled for the longest time [25].

2.4 Multi Objective Optimisation

2.4.1 Multi Objective Optimisation Algorithms

In single-objective optimisation, candidate solutions can be totally ordered with respect to a single measurement, so that any pair of solutions can be classified as better, worse, or equal. For MOO this is usually not the case. Targets are often conflicting, so that a gain in one target is very often accompanied by a loss in another target. This makes it difficult to determine that one solution is strictly better than another and often reduces the choice to a subjective preference regarding which trade-off is considered best. Instead, optimisation problems can seek to approximate a set of nondominated solutions that form the PF, which is the set of

solutions representing the optimal trade-offs between the objectives. [17, pp. 182-183]

Let $\mathcal{X}$ denote the design space of decision variables and let $f : \mathcal{X} \to \mathbb{R}^m$ denote the vector-valued objective function with components $f_i : \mathcal{X} \to \mathbb{R}$, $i = 1, \ldots, m$. Throughout this work, all objectives f_i are assumed to be minimised. Pareto dominance is then defined as follows. [18]

$$\forall i \in \{1, \ldots, m\}, \quad f_i(x) \leq f_i(y) \quad \text{and} \quad \exists j \in \{1, \ldots, m\}, \quad f_j(x) < f_j(y) \quad \Rightarrow \quad x \prec y \tag{2.9}$$

Here, $x \prec y$ denotes that solution x dominates solution y, that is, x is no worse than y in all m objectives f_i and strictly better in at least one objective. The set of Pareto-optimal decision vectors in $\mathcal{X}$ is often referred to as the Pareto set, and their image under f in $\mathbb{R}^m$ forms the PF.

In order to optimise a MOO problem, several different approaches are available. One straightforward approach is to construct a single scalar objective from all objectives, as introduced in Subsection 2.3.2, and thereby convert the problem into a single-objective optimisation problem. [36] However, such scalarisation approaches depend critically on the choice of weights or transformation. Consequently, multiple optimisation runs with varying scalarisation parameters are typically required to explore different trade-offs, and the resulting set of solutions may still provide an incomplete approximation of the PF. More advanced evolutionary algorithms, such as NSGA-II [66], R-NSGA-II [67], U-NSGA-III [68], or SPEA2 [69], are based on the concept of natural selection and evolve a population of solutions from random starting points towards the PF.

In general, there are several major families of optimisation approaches that are relevant for this work:

- **Bayesian optimisation** uses probabilistic SMs to efficiently explore expensive black-box functions [61, 70, 71].
- **Evolutionary and population-based algorithms** These methods iteratively evolve a population of candidate solutions and are widely used for black-box MOO problems:
  - Particle Swarm Optimisation [72]
  - Differential Evolution [73]
  - NSGA-II [66, 74]
- **Gradient-based optimisation** These methods are efficient when gradients are available or can be reliably approximated with `L-BFGS-B` [75, 76] being a popular algorithm.

`L-BFGS-B` belongs to the family of quasi-Newton methods and extends the `limited-memory BFGS (L-BFGS)` algorithm to efficiently handle large-scale optimisation problems with simple box constraints on the variables [37, 76]. Unlike the original L-BFGS method [75], which is designed for unconstrained optimisation, `L-BFGS-B` incorporates a projection step and an active-set strategy to ensure that parameter updates remain feasible within specified bounds. In the `scikit-learn` python package, the `L-BFGS-B` solver is used as a default optimisation algorithm. In the context of MOO, such gradient-based methods are applied to a scalarised objective, where multiple targets are combined into a single differentiable value, so that the inherently single-objective `L-BFGS-B` algorithm can be used within a multi-objective optimisation framework.

NSGA-II is used in the MOO test-rig experiment, as it has already proven effective in the considered application domain, and exploring alternative algorithms is out of scope for this work. NSGA-II is an improved version of the original NSGA algorithm [77] and was proposed by Deb et al. [66]. The algorithm is based on the concept of nondominated sorting, where solutions in a population are ranked into different fronts. The first front contains solutions that are not dominated by any other solutions in the population. The second front consists of solutions that are only dominated by those in the first front, and this process continues recursively. NSGA-II enhances the original NSGA by introducing a fast $O(MN^2)$ nondominated sorting algorithm, compared to $O(MN^3)$ in the original NSGA implementation, incorporating elitism through the use of combined parent–offspring populations, and replacing the sharing function with a parameter-free crowding-distance mechanism to preserve diversity. These improvements make NSGA-II both computationally efficient and effective in finding a well-distributed and converged approximation to the true Pareto-optimal front within a limited evaluation budget.

2.4.2 Hypervolume

As introduced in Subsection 2.4.1, in MOO with conflicting targets there is typically no single optimal solution; instead, each target has its optimum at different locations, leading to a set of optimal trade-off solutions called the PF. In order to compare different optimisation results and their PFs, various metrics have been proposed in the literature. One of the most commonly used metrics is the Hypervolume (HV). It defines the area in a two-dimensional objective space, or the HV in higher-dimensional objective spaces, that is dominated by the PF and bounded by a reference point [78, 79]. A two-dimensional visualisation of the HV is shown in Fig. 2.7.

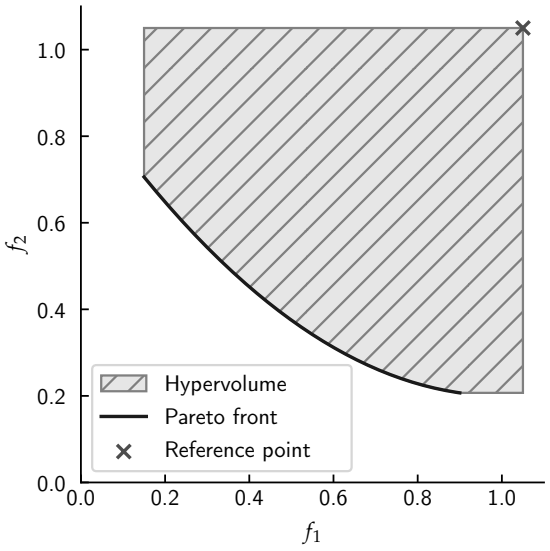


Figure 2.7: Visualisation of Hypervolume

2.5 Optimisation Test Function

This work uses the six-dimensional Hartmann-6 function as a benchmark to test and evaluate different aDoE strategies [80]. Hartmann-6 is a widely used synthetic test problem for global optimisation and SM-based design of experiments [81–83]: it is moderately high-dimensional, strongly nonlinear, and exhibits several competing local minima in addition to a single global minimum. The input space is the unit hypercube $\mathbf{x} \in [0,1]^6$, and the scalar objective combines

four anisotropic radial basis "bumps" whose locations and shapes are specified by the matrices **A** and **P**.

Following Picheny et al. [80], we use the rescaled form of the Hartmann-6 function without the additional observational noise term, defined as:

$$f(\mathbf{x}) = -\frac{1}{1.94}\left[2.58 + \sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\right)\right], \tag{2.10}$$

where

$$\boldsymbol{\alpha} = \begin{pmatrix} 1.0 \\ 1.2 \\ 3.0 \\ 3.2 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 10 & 3 & 17 & 3.50 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}, \tag{2.11}$$

and

$$\mathbf{P} = 10^{-4} \times \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}. \tag{2.12}$$

The constants 2.58 and 1.94 define an affine rescaling of the original Hartmann-6 function such that the response has approximately zero mean and unit variance over $[0,1]^6$; this preserves the location of the optimum while making the range of values more convenient for numerical experiments.

The global minimum of Hartmann-6 is given by

$$\mathbf{x}^* = (0.20169,\ 0.150011,\ 0.476874,\ 0.275332,\ 0.311652,\ 0.6573), \tag{2.13}$$

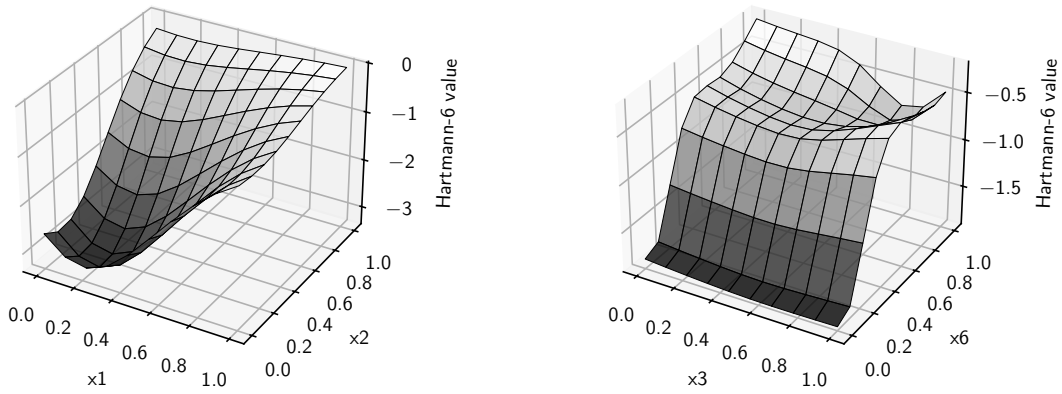
and the corresponding value of the *scaled* function is

$$f_{\text{orig}}(\mathbf{x}^*) = -3.04245774. \tag{2.14}$$

The rescaled objective (2.10) differs only by this linear transformation, so that all level sets and the optimizer $\mathbf{x}^*$ are unchanged.

From a structural point of view, Hartmann-6 can be interpreted as a weighted sum of four anisotropic Gaussian RBFs. Each row of **P** specifies the center of one basis function, the diagonal entries of **A** control how narrow or broad the basin is along each coordinate direction, and the weights α_i set the relative depths of the corresponding wells. This leads to a highly non-separable, strongly interacting response surface: some directions exhibit sharp, localized curvature, while others are comparatively flat. As a consequence, there are multiple local minima of different depths and widths, embedded in relatively low-variation regions, which makes Hartmann-6 a challenge for global search and aDoE algorithms.

Figure 2.8 illustrates two-dimensional slices of the function obtained by fixing four of the six inputs and varying the remaining two. In both panels the horizontal axes correspond to the two free input dimensions, while the color map (equivalently, the "z-axis" of the response surface) encodes the objective value. Panel 2.8a shows a slice in the (x_0, x_1)-plane that contains a single dominant valley, whereas panel 2.8b, taken in the (x_2, x_5)-plane, reveals a more corrugated landscape with several competing valleys. These slices highlight the anisotropy and strong interaction structure of Hartmann-6: depending on which subset of variables is inspected, the problem can appear either relatively benign or highly multimodal, which is precisely the kind of behaviour that stresses adaptive design and optimisation strategies.



(a) Slice with free dimensions x_1 and x_2.

(b) Slice with free dimensions x_3 and x_6.

Figure 2.8: Two-dimensional slices of the Hartmann-6 function with four inputs fixed and two varied. The colour map (z-axis) indicates the objective value.

2.6 Analysis of Variance

The permutation-based Analysis of Variance (ANOVA) procedures used in this work follow the general framework for permutation tests in multi-factorial ANOVA proposed by Anderson et al. [84]. In particular, permutation tests are constructed by identifying the exchangeable units implied by the denominator mean square of the relevant F-ratio and by generating an empirical null distribution of a classical ANOVA test statistic through repeated random reallocation of these units.

ANOVA provides a general inferential framework for assessing whether the mean values of a response variable differ across multiple groups. Originally introduced by Fisher [19], classical ANOVA evaluates whether the observed between-group variability, known as the sum of squares between (SSB), exceeds what would be expected under the null hypothesis of equal population means. In its standard form, ANOVA partitions the total variability of a dataset into components attributable to systematic group differences and residual noise.

In this work, only one-way between-groups ANOVA designs are used, where the factor represents either an acquisition strategy or a particular parameter setting of a strategy. The focus is on testing for differences in continuous performance metrics (e.g. Gap to Optimum, global and local RMSE) across these groups.

Let Y_{ij} denote the j-th observation from the i-th group, for $i = 1, \ldots, a$ and $j = 1, \ldots, n_i$, with group means $\bar{Y}_i$, overall mean $\bar{Y}$, and group sizes n_i. Here, a denotes the number of groups, and $N = \sum_{i=1}^{a} n_i$ is the total sample size. The total sum of squares decomposes as

$$\text{SST} = \underbrace{\sum_{i=1}^{a} n_i (\bar{Y}_i - \bar{Y})^2}_{\text{SSB}} + \underbrace{\sum_{i=1}^{a} \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_i)^2}_{\text{SSW}},$$

where sumofsquaresbetween(SSB) captures between-group differences and sumofsquareswithin(SSW) describes within-group variability. Classical ANOVA converts these into mean squares and

constructs the F-statistic

$$F = \frac{\text{SSB}/(a-1)}{\text{SSW}/(N-a)},$$

Under normality and homoscedasticity assumptions, F follows an F-distribution and significance is assessed by comparing the observed F-value to its null distribution.

The validity of the classical F-test hinges on three key assumptions: (i) independence of observations; (ii) approximately normally distributed residuals; (iii) homogeneity of variances across groups. In situations, such as the one examined in this work, where these assumptions cannot be met, a permutation-based ANOVA can be used to obtain valid inference without relying on parametric distributional results.

A non-parametric, permutation-based variant of one-way ANOVA retains the same effect size as classical ANOVA but constructs the null distribution empirically. Let T denote the classical one-way ANOVA F-statistic computed from the original data, $T_{\text{obs}} = F_{\text{obs}}$. Under the null hypothesis that group labels are unrelated to the response, the joint distribution of the data is invariant to permutations of the labels. Permutation tests exploit this by repeatedly reshuffling group labels among the observations and recomputing the test statistic. For each permutation, the statistic

$$T_{\text{perm}} = F_{\text{perm}},$$

is recomputed from the permuted dataset. The permutation p-value is then estimated as

$$p_{\text{perm}} = \frac{\#\{T_{\text{perm}} \geq T_{\text{obs}}\} + 1}{N_{\text{perm}} + 1},$$

which quantifies how likely the observed separation of group means is under the null hypothesis of no systematic differences. Because this test relies only on the exchangeability of observations under the null, it avoids explicit normality assumptions and is typically more robust to non-Gaussian residuals, unequal group sizes, and moderate heteroscedasticity.

Permutation ANOVA thus provides a principled and assumption-light hypothesis test while still allowing quantification of statistical significance.

When the omnibus permutation test indicates significant group differences, specific contrasts are evaluated using permutation-based pairwise tests. For two groups A and B, with sample means $\bar{Y}_A$ and $\bar{Y}_B$, the test statistic is taken as the difference in means

$$T_{\text{obs}} = \bar{Y}_A - \bar{Y}_B,$$

and a two-sided test is obtained by comparing $|T_{\text{obs}}|$ to the permutation distribution of $|T_{\text{perm}}|$ generated by repeatedly permuting the group labels between A and B. This yields a non-parametric p-value for the contrast. To control the family-wise error rate across multiple pairwise comparisons, the Holm step-down procedure [85] is applied to the set of unadjusted p-values.

Although the computational procedure differs, permutation ANOVA retains the conceptual structure of classical ANOVA. An effect size derived from between-group variability is combined with a null distribution describing random variation, and a p-value characterises the strength of evidence against the null hypothesis. The key difference is that the null distribution is constructed empirically by permutation rather than analytically from parametric assumptions, allowing rigorous inference even when the classical assumptions do not hold.

2.7 Related Work

ADoE originated from the need to optimize expensive black-box functions with minimal evaluations. A seminal contribution was the single objective Efficient Global Optimisation (EGO) algorithm, which introduced a Bayesian sequential sampling strategy using a EI AF [64]. This laid the groundwork for later methods that could iteratively choose new experiment points based on current model predictions and uncertainty. Building on this, Knowles proposed the ParEGO algorithm to address multi-objective problems by aggregating multiple objectives into a single weighted scalar function and then applying EGO-style optimisation [36]. ParEGO suggests an apporach which balances exploration and exploitation across trade-offs and has shown superior performance compared to the state-of-the-art at the time on several benchmark tasks at that time.

In the automotive domain, research on aDoE began gaining traction in the early 2010s. Hartmann and Nelles introduced an adaptive test planning method for engine calibration based on Hierarchical Local Model Tree (HiLoMoT), which suggested not using one global SM but many localised SMs instead. The design space is partitioned based on local model quality, and additional samples are placed in regions where the model exhibits low accuracy [99]. Klein et al. applied this HiLoMoT-DoE strategy in a real-world engine calibration task, demonstrating that adaptive measurements can reduce test bench time while maintaining modelling quality [86]. These efforts were among the first to demonstrate the benefits of aDoEs for internal combustion engine calibration, a domain with complex, nonlinear behaviour and high cost per measurement. However, as the approach was released as a MATLAB toolbox, it could not be reused directly in this work, and a full reimplementation in python was beyond the scope of this work.

Subsequent work extended these concepts to multi-objective drivability calibration tasks. Dursun et al. experimented with aDoE for drivability optimisation, using a Round-Robin strategy to alternate among multiple regression models representing different objectives [25]. This approach allowed a more balanced learning process and improved model accuracy compared to purely sequential or static designs. Tietze further advanced the field by incorporating GPR for model-based calibration, leveraging GPR's uncertainty estimates to guide the experiment selection process [97]. GPR became increasingly popular in automotive aDoE contexts due to its ability to capture both predictions and uncertainty estimates effectively.

In recent decades, researchers in the machine learning community additionally made advances in tackling the challenges of MOO aDoE. Zhang et al. proposed a framework for selecting not only where to sample but also which objective to query, aiming to minimize predictive uncertainty across all targets using muti-objective Gaussian processes [87]. These ideas resonated with automotive calibration, where multiple drivability or emissions outputs must often be optimized together. In this vein, Prochaska et al. developed the Active Output Selection strategy to prioritize experiments based on cross-validation error of competing SMs [103]. Their results demonstrated that adaptively focusing on the most uncertain model at each iteration could reduce the number of required measurements by up to 30 % compared to traditional Round-Robin or global approaches. This technique marked a step forward in robust aDoE for mutiobjective regression settings and has been validated on both benchmark functions and automotive test data.

Complementary research also focused on improving the robustness of adaptive learning strategies under real-world noise. Park and Kim introduced a robust expected model change criterion to select experiments that not only promise improvement but are less sensitive to outliers and uncertainty misestimation [58]. This is especially relevant in physical testing environments like engine calibration, where measurements may be noisy or difficult to reproduce precisely. Recently, Zhu et al. demonstrated a concurrent Bayesian optimisation framework for multi-condition engine calibration, combining evolutionary strategies with probabilistic modeling to select optimal test points across several engine speed/load regimes simultaneously [88]. This concurrent approach exemplifies how modern aDoE methods can scale to high-dimensional, multi-objective industrial problems while delivering significant gains in data efficiency.

Overall, the research trajectory in adaptive design of experiments, from foundational single-objective strategies [64], through multi-objective extensions [36, 87], to robust and application-ready frameworks [25, 103, 88], reflects the increasing maturity of this methodology within automotive calibration contexts.

The practical context of this work comes with constraints arising from the limited and costly availability of test-rig time. Whenever time on the test-rig is allocated, the corresponding experiments must therefore be highly time-efficient and reliable in order to make best use of the scarce experimental resources, an aspect that is only implicitly addressed in much of the work discussed above. Many existing aDoE approaches primarily focus on minimising the number of samples or function evaluations, without explicitly accounting for the often substantial and heterogeneous time required both to obtain individual measurements and to update the respective SMs. This is particularly critical in sequential frameworks, where model fitting and trial execution are performed serially rather than in parallel or even asynchronous as suggested in this work.

However, most prior work on aDoE has predominantly focused on reducing the number of measurements, under the assumption that each sample represents the primary cost driver, and therefore often neglects time efficiency as a distinct objective. This is particularly valid in engine calibration, where poor parameter combinations may lead to mechanical wear or damage. In contrast, drivability calibration presents different constraints: here, mechanical wear is typically less critical, and the dominant cost factor is the usage time of the test-rig. Consequently, in the present work a key optimisation objective also becomes the minimisation of the total experiment duration rather than the sample count alone.

3 Simulative Experiment

This Chapter particularly deals with the question how different pool-based aDoE strategies compare in terms of optimisation performance and SM quality on a non-trivial benchmark function. A special focus is put on the `ADEI` strategy as it was implemented into the test-rig experiment prior to the simulative study following below.

Section 3.1 describes the setup and methodology of the simulative experiments, while Section 3.2 presents and discusses the results obtained.

3.1 Setup and Methodology

The simulative aDoE setup serves to examine the behaviour of different sampling strategies in a controlled environment. Simulation with inexpensive test functions allows many repetitions of different strategies and enables comparison of the results to the true optimum of the test function, which is not possible for the real-world drivability optimisation task where the true optimum is unknown and the test-rig resources are limited.

The goal of aDoE is to use the knowledge gained during experimentation to guide the remaining samples within the available budget towards regions of interest. Even though the real-world scenario poses a MOO problem, the simulative experiments are carried out in a single-objective setting in order to reduce complexity and allow for more repetitions within the available time frame. This is justified because, in the MOO case, one SM per target is built, which, when considered individually, can be reduced to a single-objective regression task. When dealing with a MOO problem, the AF can be aggregated into one scalar, or the experiments can be carried out in a Round-Robin manner, as previously mentioned in Subsection 2.3.2.

3.1.1 Surrogate Model Setup

As outlined in Subsection 2.1.1, the workflow currently employed for drivability optimisation uses GPR models, which were introduced in Section 2.2.

Since they have been shown to perform well in engine and drivability optimisation [26, 27, 103, 89] as well as in other applications [90, 91] this architecture was selected as the SM architecture for the following aDoE experiments as well.

The MATLAB-based MB internal tool fits models using the default `ARDRationalQuadratic` kernel. ARD allows the kernel to employ anisotropic length-scales, i.e. one length-scale per input dimension, which can improve model quality and provide an indicator of feature importance, with larger length-scales indicating lower relevance of an input and smaller length-scales indicating higher relevance.

The scikit-learn implementation in python also offers a `Rational Quadratic` kernel, but without ARD support. Therefore, in order to benefit from anisotropic length scales, a `Matérn` kernel with ARD enabled and smoothness parameter $\nu = 0.5$ is used instead. A `Matérn` kernel with $\nu = 0.5$ corresponds to an exponential covariance function and is suitable for modelling comparatively rough response surfaces.

Unless stated otherwise, the GPR models are configured as follows. The scikit-learn GPR im-

plementation is used with a combined kernel consisting of a `ConstantKernel` multiplied by a `Matérn` kernel with ARD and $\nu = 0.5$. Additionally a `WhiteKernel` is added to account for observation noise. Three optimizer restarts are performed for the kernel hyperparameters in order to reduce the risk of poor local optima while keeping the computational effort acceptable. In each model update, the optimized kernel from the previous iteration is used as the starting point for the HPO, which stabilises the fit over successive adaptive steps. This fixed kernel architecture is intentionally restrictive: it does not target the best possible model for the Hartmann-6D function but aims to provide a robust SM that can be fitted quickly across many repetitions to ensure the design plan can be updated frequently. When the aDoE process is completed, it is therefore recommended to refit a final SM using more flexible kernels and more extensive HPO to yield the best possible model quality.

Furthermore, the aDoE framework implemented in this work explicitly relies on open-source tooling as an alternative to proprietary platforms such as MATLAB, which were frequently employed in earlier studies. The entire framework is realised in python 3.10.18, which introduces both opportunities and constraints. For instance, while python offers flexibility and transparency, some established algorithms such as HiLoMoT are not readily available and are non-trivial to implement. Nevertheless, this open-source approach facilitates reproducibility, extensibility, and cost efficiency and aligns with current trends towards more accessible and modular research workflows. In summary, the choice of an open-source toolchain is an integral part of the practical contribution of this thesis to advancing aDoE for drivability calibration under real test-bench conditions.

3.1.2 Setup of Sampling Strategies

Experimental Design

A central component of any aDoE procedure is the AF, as introduced in Subsection 2.3.2. This section describes the sampling strategies evaluated in this work and the experimental design used to compare them. All strategies are benchmarked on the scaled Hartmann-6 function introduced in Section 2.5.

Each strategy is tested under various experimental settings. The total sampling budget is defined by `n_max`, with the first `n_init` points generated via LHS to form the initial DoE. Adaptive sampling then proceeds in iterations, where in each iteration the `samples_per_it` highest-scoring candidates according to the current AF are selected from the remaining candidate pool.

All aDoE strategies are implemented as pool-based samplers. At the start of each run, a single candidate pool of size $\texttt{n\_max} \times \texttt{osfac}$, with `osfac` being the oversampling factor, is generated using LHS.

Selected points are removed from this pool as sampling progresses. The exploitation ratio `exrat` controls the number of samples obtained through direct SM optimisation. After collecting $\texttt{n\_max} \times (1 - \texttt{exrat})$ measurements, the SM is refitted in each subsequent iteration and the next point is obtained by applying `sklearn`'s `minimize` method to the SM which runs the `L-BFGS-B` algorithm. This process continues until the full budget `n_max` is reached.

The experimental factors and their investigated levels are summarised in Table 3.1.

Due to the prohibitively large number of possible combinations of the five parameters listed in Table 3.1, amounting to 1 296 configurations, only a subset of parameter combinations is

Table 3.1: Experimental parameters and values tested in the preliminary experiments.

Parameter name	Values tested
n_max	[100, 200, 500, 1000]
n_init	[10, 50, 100]
osfac	[2, 100, 1000]
exrat	[0.0, 0.05, 0.1, 0.2]
samples_per_it	[1, 5, 10, 20]
af	[US, EI, UCB]

evaluated. As the default setting, the baseline configuration provided in Table 3.2 is used.

Table 3.2: Baseline experimental parameters for simulative aDoE.

	n_max	n_init	osfac	exrat	samples_per_it
Baseline	200	50	1000	0.0	5

Using this baseline configuration, the influence of each parameter is examined individually by varying one parameter at a time while keeping all others fixed. The purpose of this simulative study is explicitly exploratory. The aim is not to determine a configuration that is optimally tuned to the Hartmann-6 function, but to obtain an initial understanding of which strategy parameters have a pronounced influence on performance and where robust regions of the parameter space may lie. Accordingly, the subsequent statistical analysis is interpreted as providing guidance on sensitivity and qualitative trends rather than prescriptive recommendations for a specific benchmark problem.

As a reference, a purely space-filling LHS design, denoted `Random`, is included. In this baseline, `n_max` samples are generated using LHS, and the parameters `n_init`, `osfac` and `samples_per_it` do not have an effect and therefor are ignored.

To account for stochastic variability, each combination of strategy and parameter setting is executed twenty times with different seeds. For a given run ID, all strategies share the same random seed, so that the initial LHS design and the candidate pool are identical across strategies. The only exceptions are `Random` and `SemiAD`, which both have the same pool size equal to `n_max`, with `SemiAD` and `Random` differing only in the order in which samples are measured. This common-random-numbers design improves comparability of strategies within a run by reducing variance due to different initial designs and candidate pools [20]. The resulting twenty runs per strategy and configuration are treated as independent replicates in the subsequent statistical analysis.

Sampling Strategies

The following sampling strategies are considered in this work. They instantiate the acquisition functions presented in Subsection 2.3.2 for the Hartmann-6 benchmark function defined in Section 2.5.

LHS (`Random`): This non-adaptive baseline is purely explorative. It employs LHS sampling across the full budget `n_max` without adaptive updates and does not use any acquisition function.

Semi-adaptive design (`SemiAD`): `SemiAD` is a adaptive variant of the random baseline. It uses a pool-based approach similar to the other strategies, with the key difference that its pool size equals the budget `n_max`. Consequently, when using all samples for model fitting, `SemiAD` and `Random` use the same training data, given the same seed was used for the initial LHS design. This strategy is used to analyse whether model quality can be improved with fewer data, making a case for early-stopping procedures that would prematurely end the experiment before reaching the full budget when no further improvement is observed in the last few iterations, thereby avoiding unnecessary measurements.

Adaptive uncertainty sampling (`ADUS`): `ADUS` is a purely explorative strategy based on US. Its AF is the predicted Standard Deviation (STD) of the GPR model, corresponding to the US AF introduced in Subsection 2.3.2. In each iteration, the samples with the highest predicted uncertainty are selected from the candidate pool. The underlying assumption is that regions of high predicted uncertainty indicate where the model can be improved most, such that this strategy may provide a fast route to improving overall model quality.

Expected-improvement sampling (`ADEI`): The strategy `ADEI` uses Expected Improvement (EI) as AF. Here, the utility of a candidate is defined by its EI over the current best observed value, as introduced in Subsection 2.3.2, and a larger value implies a more desirable candidate. This strategy balances exploration and exploitation by considering both the predicted mean and uncertainty of the GPR model, with a stronger emphasis on exploitation initially as described in Subsection 2.3.2. When candidates are close to the previous best observed value, the STD has a greater influence on the EI value, promoting exploration.

Upper-confidence-bound sampling (`ADUCB`): The strategy `ADUCB` employs the Upper Confidence Bound (UCB) AF. This approach is similar in spirit to EI but explicitly controls the trade-off between exploration and exploitation through the exploration weight κ introduced in Subsection 2.3.2. Depending on the choice of κ, `ADUCB` can be configured to behave more exploratively or more exploitatively. This work uses a κ value of 2, which is a commonly used default value in the literature and frameworks such as *scikit-optimize*[1] and *GPyOpt*[2].

3.1.3 Evaluation Methodology

Model performance is assessed on an independent validation set consisting of 1 000 LHS-sampled points from the input space. Global model quality metrics are computed using the full validation dataset, while additional local metrics are evaluated for 5, 10, and 25 % of the validation data closest to the true global optimum. In particular, the metrics considered are the Gap to Optimum, the global RMSE, and the RMSE using 5 % of the validation data closest to the true global optimum. The local metrics allow to evaluate whether model quality near the optimum deviates from the global model quality and hence might be favourable for optimisation tasks.

To characterise the behaviour of the generated datasets over the course of the experiment, metric trajectories are constructed. For a given experiment, the dataset is taken in the order in which samples were collected and models are iteratively fitted, starting with 20 samples and then incrementally adding subsequent batches of 20 samples to the training data until `n_max` samples are reached. Each model is validated on the same validation dataset, and the resulting

[1] `https://scikit-optimize.github.io/0.8/modules/generated/skopt.optimizer.base_minimize.html?highlight=upper%20confidence` (visited on 01.12.2025)

[2] `https://gpyopt.readthedocs.io/en/latest/GPyOpt.acquisitions.html#module-GPyOpt.acquisitions.LCB` (visited on 01.12.2025)

metrics are plotted against the number of samples used to train the model, yielding a metric trajectory that illustrates how model performance evolves as additional data become available.

A complementary perspective focuses on optimisation performance. Here, the final SM trained on the full dataset is used to perform an optimisation using the default method of the `minimize` routine from the `scipy.optimize` module. The solution obtained is then evaluated on the true Hartmann-6 function and compared against its known global optimum, providing a practical measure of the model's suitability for optimisation tasks.

Before comparing strategies, the suitability of a classical parametric one-way ANOVA was examined. The normality of residuals was assessed using the Shapiro–Wilk test, and the homogeneity of variances was examined using Levene's test. The results of these assumption checks, summarised in Table A.1, indicate that the assumptions of normality and homoscedasticity are not consistently satisfied across all metrics.

Consequently, permutation-based ANOVA is used as the primary inferential method for all subsequent analyses. Following the framework introduced in Section 2.6, for each parameter and metric a global permutation ANOVA at a significance level of $\alpha = 0.05$ is conducted to test for overall differences between groups, i.e., strategies or parameter settings. For pairwise permutation tests, Holm correction is applied to compensate for multiple testing and to identify which groups differ. Inference and conclusions are based on permutation tests with 100 000 resamplings to obtain accurate p-value approximations.

3.2 Discussion of Simulation Experiment Results

In this Section several ANOVA-based analyses are presented to compare the significance of differences between the sampling strategies and parameter settings. These ANOVA-tables include p-values which indicates the probability of observing the difference between the groups if the null hypothesis were true as introduced in Section 2.6. These p-values are annotaed with asterisks according to the following scheme to improve readablility: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

3.2.1 Baseline Configuration: Comparison of Strategies

In this subsection, we first focus on the Gap to Optimum as the primary performance metric and subsequently analyse the global and local RMSE to assess model quality.

The one-way ANOVA results for the default configuration and all metrics comparing the different strategies are presented in Table 3.3. For each metric, at least one strategy differs significantly from the others. As the aim of this work is to find a strategy to facilitate optimisation, the pairwise comparison of the Gap to Optimum metric, shown in Table 3.4, is particularly relevant.

Analysis Gap to Optimum

The pairwise tests in Table 3.4 indicate that the difference in performance between `ADEI` and `ADUCB` is not statistically significant, whereas both differ significantly from the remaining strategies. The mean values of the Gap to Optimum for the different strategies in Table 3.5 show that `ADEI` and `ADUCB` outperform the remaining strategies with respect to this metric. It can be seen that the mean Gap to Optimum decreases from 0.2858 for `Random` and `SemiAD` to 0.0607 and below for `ADEI` and `ADUCB`, which corresponds to an improvement of nearly 79 %. Considering the STD, `ADUCB` not only has the lowest mean but also the lowest STD of all strategies, with

Table 3.3: Global one-way permutation ANOVA for `strategy` across metrics using the default values of Table 3.2.

metric	F_perm	p_perm
gap_to_optimum	27.699	0.000***
rmse	54.966	0.000***
rmse_near50p	12.442	0.000***
rmse_near25p	5.513	0.000***
rmse_near10p	3.796	0.006**
rmse_near5p	4.682	0.002**

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$

Table 3.4: Pairwise permutation comparisons for `strategy` on metric `gap_to_optimum` (configuration: n_max=200, n_init=50, osfac=1 000, exrat=0.0, samples_per_it=5).

group1	group2	diff_mean	p_adj_Holm
ADEI	ADUCB	0.016	0.261
ADEI	ADUS	-0.421	0.000***
ADEI	Random	-0.253	0.000***
ADEI	SemiAD	-0.253	0.000***
ADUCB	ADUS	-0.436	0.000***
ADUCB	Random	-0.269	0.000***
ADUCB	SemiAD	-0.269	0.000***
ADUS	Random	0.167	0.085
ADUS	SemiAD	0.167	0.085
Random	SemiAD	-0.000	1.000

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$

0.031 being only a quarter of the STD of `ADEI`. Additionally, the pairwise tests for the Gap to Optimum suggest that the strategies can be grouped into two sets that, within each group, show no statistically significant differences. One group is formed by `Random`, `SemiAD`, and `ADUS`, and the other by `ADEI` and `ADUCB`.

To understand why `SemiAD` and `Random` are indistinguishable in all metrics, note that the `SemiAD` approach is equivalent to the `ADUS` strategy with an `osfac` of 1. It is therefore not surprising that `SemiAD` and `Random` show identical performance in all experiments, as they both use the same training data. The only difference is the order in which the data are collected, which does not influence the final model but only the trajectory of the metrics. However, in the implementation used here, the order does not seem to affect the trajectories significantly, as can be seen in Figure 3.1, where the two trajectories are almost congruent for both the global RMSE as well as for the localized RMSE. Possible reasons for this behaviour are discussed later in Chapter 5.

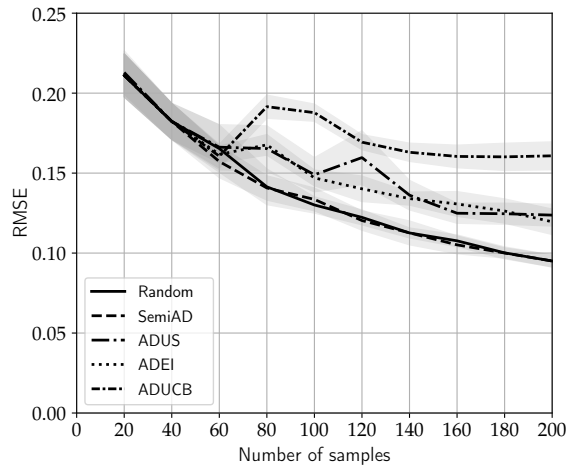
Analysis Global and Local RMSE

As indicated by the one-way permutation ANOVA in Table 3.3, the strategies differ significantly in RMSE. Analysing global and local RMSE reveals further insights into their behaviour. Applying an aDoE is expected to improve model quality, especially near the optimum.

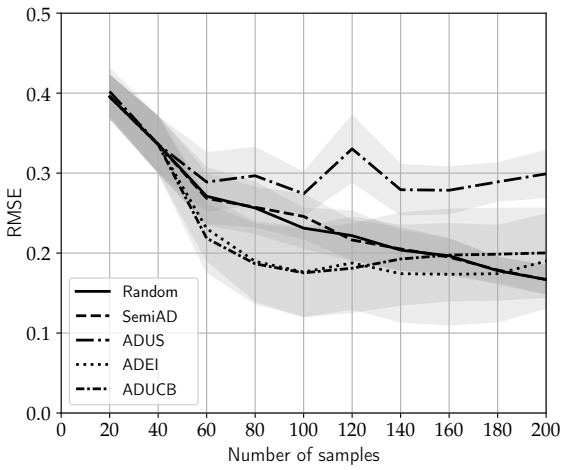
Table 3.5: Summary statistics of gap to true optimum for each strategy.

strategy	mean	median	std
ADEI	0.060700	0.005100	0.121800
ADUCB	0.024900	0.001000	0.030700
ADUS	0.349900	0.276200	0.259600
Random	0.285800	0.266700	0.148800

From the trajectories in Figure 3.1, it becomes apparent that the strategies behave very differently from a global and a local perspective. For the global RMSE trajectories in Figure 3.1a, the baseline strategies `Random` and `SemiAD` consistently outperform the adaptive strategies and are nearly indistinguishable from each other. `ADUS` and `ADEI` perform more than 25 % worse than `Random` and `SemiAD` at 200 samples, with `ADUCB` lagging even further behind by more than 68 % relative to `Random`. Overall, `Random` and `SemiAD` exhibit the fastest improvements in global model quality among all strategies.



(a) Global RMSE trajectory using 100 % validation data.



(b) Local RMSE trajectory with 5 % of the validation data closest to the real optimum.

Figure 3.1: Metric trajectories for the default configuration of the aDoE simulation with 95 % confidence intervals. (Data excerpt in Table A.2.)

This picture changes when considering Figure 3.1b, which presents the RMSE computed only on the 5 % of validation points closest to the true optimum, in this case amounting to 50 samples. Here, `ADEI` and `ADUCB` exhibit the steepest initial improvement and outperform the other strategies, plateauing after approximately 80 samples. Again, no distinction is visible between `Random` and `SemiAD`, both of which start at lower performance than `ADEI` and `ADUCB` but converge to similar model quality after 200 samples. In this local perspective, `ADUS` performs worst among all strategies, showing little improvement after the initial test plan.

The most notable observation is that `ADEI` and `ADUS` appear similar when judged by the global RMSE, whereas for the local RMSE, `ADUS` appears to perform much worse, and instead `ADUCB` seems to catch up and performs similarly to `ADEI`. Based solely on this information, `ADEI` seems to provide a good trade-off between local and global metrics. A reason for this could be, that

the AF EI is stopping to exploit when the pool does not contain points that are significantly better than the current best observation one. UCB on the other hand has no such intrinsic stopping criteria and continues to exploit the current best regions of the model which might not be desirable and cause a drop in global model quality as is observed in this experiment. Taken together, these results indicate that `ADEI` and `ADUCB` are the most effective strategies in terms of optimisation performance, with `ADEI` offering a compromise between local and global model quality. Furthermore, it shows that an AFs with an exploitative character indeed are beneficial for optimisation tasks even when models are trained with little data.

Summarizing the performance of all strategies using the default configuration, it can be said that the pool-based adaptive strategies `ADUS`, `ADEI`, and `ADUCB` deviate significantly from the `Random` baseline. When it comes to minimizing the Gap to Optimum for the Hartmann-6 function, `ADEI` and `ADUCB` outperform the baseline, while in terms of global model quality the baseline cannot be beaten or matched by any adaptive strategy. `ADUS`, however, consistently lags behind the baseline across all metrics. `ADEI` seems to provide a favourable compromise between optimisation performance and model quality, excelling in reducing the Gap to Optimum while achieving competitive global and local RMSE values.

3.2.2 Sensitivity of ADEI to Strategy Parameters

After considering the results of the default configuration, we now investigate, as an example, how changes in the parameters listed in Table 3.1 affect the performance of the `ADEI` strategy. In the following, only `ADEI` is analysed, as it serves as a representative strategy that provides a favourable compromise between optimisation performance and model quality compared to the other strategies. In particular, we first investigate parameters related to the sample budget and update frequency (`n_max`, `samples_per_it`, `n_init`) and then parameters that shape the acquisition process (`osfac`, `exrat`).

Effect of the Sample Budget `n_max`

First, `n_max` is varied while keeping all other parameters at their default values using the values 100, 200, 500 and 1 000. Table 3.6 shows the results of the one-way permutation ANOVA for all metrics when varying `n_max` for the `ADEI` strategy. Significant differences can be observed across all metrics. As the Gap to Optimum is particularly relevant for the practical context of this work, only the pairwise comparisons for this metric are discussed in detail below.

The box plots in Figure 3.2 show the Gap to Optimum for different `n_max` values. While the median Gap to Optimum value already is close to zero for small values for `n_max` the STD decreases a lot with increasing budget, as can be seen in Table 3.8. According to the pairwise permutation tests in Table 3.7, only the differences between `n_max` values of 100 and 1 000, as well as 200 and 1 000, are statistically significant.

Generally, the choice of `n_max` is critical for an aDoE framework using a GPR as SM, because the computational complexity of GPR training increases cubically with the number of samples. As `n_max` grows, model-fitting times become longer, while the measurement process on the test-rig continues at an approximately constant rate. In case of an asynchronous setup, which later is used for the real-world scenario, measurements are still being acquired while the model is being updated. This remains acceptable as long as a single model update accounts for only a small fraction of the overall sampling budget (for example, roughly the time required to acquire a few tens of additional measurements in an experiment with several hundred or a thousand samples). However, if one model update takes as long as acquiring a sizeable portion

Table 3.6: Global one-way permutation ANOVA for `n_max`

metric	F_perm	p_perm
gap_to_optimum	2.172	0.013*
rmse_near5p	5.748	0.001**
rmse	112.547	0.000***
rmse_near10p	5.727	0.001**
rmse_near25p	7.370	0.000***
rmse_near50p	15.448	0.000***

$^{*}\,p < 0.05$; $^{**}\,p < 0.01$; $^{***}\,p < 0.001$

of the total budget, the adaptive element is effectively undermined: a large number of new data points are collected based on an outdated model before the updated SM can influence subsequent sampling decisions. In summary, for `ADEI` the data shows a trend towards better optimisation performance with larger budgets which is expected.

Table 3.7: Pairwise permutation comparisons of `n_max` levels on metric `gap_to_optimum` (strategy=`ADEI`, n_max=200, osfac=1 000, exrat=0.0, samples_per_it=5).

group1	group2	diff_mean	p_adj_Holm
100	200	0.068	0.296
100	500	0.084	0.189
100	1 000	0.097	0.007**
200	500	0.016	0.269
200	1 000	0.029	0.007**
500	1 000	0.013	0.189

$^{*}\,p < 0.05$; $^{**}\,p < 0.01$; $^{***}\,p < 0.001$

Table 3.8: ADEI Summary statistics (mean, median, std) for metric `gap_to_optimum` by `n_max`.

n_max	mean	median	std
100	0.101	0.008	0.258
200	0.033	0.009	0.036
500	0.016	0.003	0.029
1 000	0.004	0.001	0.014

Effect of the Number of Samples per Iteration `samples_per_it`

The next parameter investigated is the number of samples measured between model updates, `samples_per_it`. All strategies were run with 1, 5, 10, and 20 samples per iteration while keeping all other parameters at their default values. While this is not a parameter that can be controlled during the experiment on the test-rig, as it uses an asynchronous experiment and sampling framework where this is dependent on the time taken for model fitting and data acquisition, it is still interesting to see how it affects the performance of the `ADEI` strategy in the simulative experiment. It could show that more frequent model updates lead to better performance, which might motivate efforts to speed up the process of model fitting. The one-way permutation ANOVA results in Table 3.9 indicate statistically significant differences only

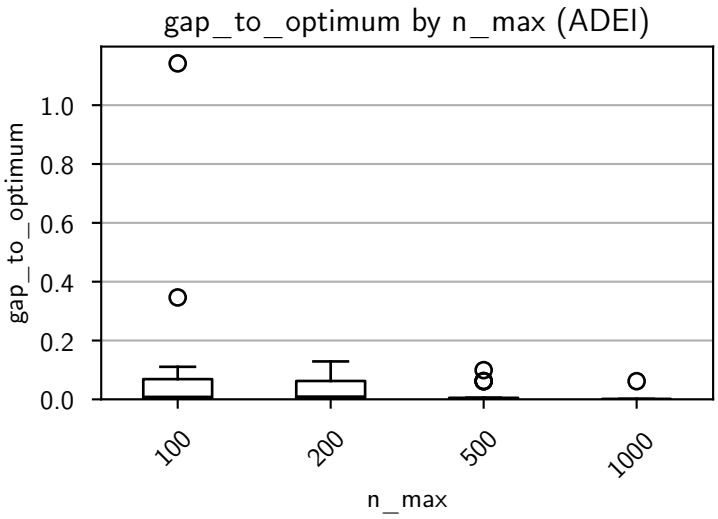


Figure 3.2: Gap to Optimum for different `n_max`. The data can be seen in Table 3.8.

Table 3.9: Global one-way permutation ANOVA for `samples_per_it`

metric	F_perm	p_perm
gap_to_optimum	0.884	0.405
rmse_near5p	0.856	0.465
rmse	3.727	0.014*
rmse_near10p	1.121	0.343
rmse_near25p	1.226	0.305
rmse_near50p	0.911	0.439

$^{*}\ p < 0.05$; $^{**}\ p < 0.01$; $^{***}\ p < 0.001$

for the global RMSE metrics when varying `samples_per_it`. All other p-values are 0.3 or higher and are thus statistically nearly indistinguishable.

A closer look at the pairwise comparison of the global RMSE in Table 3.10 indicates that a significant effect can only be observed between 5 and 20 samples per iteration, with 5 samples per iteration consistently leading to lower RMSE values. Hence, this parameter is the only one investigated here for which a lower value seems to be beneficial for the model quality, while not having the adverse effect on the Gap to Optimum observed with the other parameters. In practice, this suggests that more frequent model updates are advantageous for `ADEI`, as long as update times remain acceptable relative to measurement times, leading to the conclusion that efforts to speed up model fitting and acquisition function optimisation might be worthwhile.

Effect of the Initial Design Size `n_init`

Another interesting adjusting parameter is the size of the initial design `n_init` which is varied here between 10, 50, and 100 while keeping all other parameters at their default values. The one-way permutation ANOVA results in Table 3.11 indicate statistically significant differences for the RMSE metrics when varying `n_init`, but not for the Gap to Optimum. The pairwise comparison of the global RMSE indicates that a significant positive effect can be observed by increasing `n_init` from 10 to 50, but further increasing it to 100 does not yield significant im-

Table 3.10: Pairwise permutation comparisons of `samples_per_it` levels on metric `rmse` (strategy=`ADEI`, n_max=200, osfac=1000, exrat=0.0, samples_per_it=5).

group1	group2	diff_mean	p_adj_Holm
1	5	0.009	0.532
1	10	-0.001	0.941
1	20	-0.014	0.358
5	10	-0.010	0.358
5	20	-0.023	0.004**
10	20	-0.013	0.102

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$

Table 3.11: Global one-way permutation ANOVA for `n_init`

metric	F_perm	p_perm
gap_to_optimum	2.026	0.142
rmse	11.930	0.000***
rmse_near50p	6.474	0.004**
rmse_near25p	5.903	0.006**
rmse_near10p	4.261	0.020*
rmse_near5p	3.411	0.042*

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$

provements, as can be seen in Table 3.12. The negative value for the comparison of 10 to 50 also indicates that the RMSE decreases when increasing `n_init` to 50 but stays roughly the same when further increasing it to 100.

Table 3.12: Pairwise permutation comparisons of `n_init` levels on metric `rmse` (strategy=`ADEI`, n_max=200, osfac=1000, exrat=0.0, samples_per_it=5).

group1	group2	diff_mean	p_adj_Holm
10	50	0.041	0.001**
10	100	0.040	0.003**
50	100	-0.001	0.946

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$

In summary, for `ADEI` an initial design size of about 50 samples appears sufficient to achieve good global model quality, and substantially larger initial designs do not provide meaningful additional benefits.

Effect of the Pool Size `osfac`

We next investigate the influence of the pool size by varying `osfac` while keeping all other parameters at their default values using the values 2, 100, and 1 000. This analysis allows us to assess whether the size of the candidate pool in the aDoE strategy is critical for performance. The initial reasoning is that a larger pool provides more options for selecting promising candidates and gives more freedom for the AF. A first look at the one-way permutation AN-

Table 3.13: Global one-way permutation ANOVA for `osfac`

metric	F_perm	p_perm
gap_to_optimum	15.284	0.000***
rmse	23.902	0.000***
rmse_near50p	14.889	0.000***
rmse_near25p	7.425	0.001**
rmse_near10p	3.270	0.043*
rmse_near5p	1.786	0.174

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$

OVA results in Table 3.13 indicates statistically significant differences for all metrics except the local RMSE on 5 % of the validation data. Table 3.14 shows that all pairwise comparisons for the Gap to Optimum metric are statistically significant. Further, the `diff_mean` between the groups shown in Table 3.14 illustrate that increasing `osfac` leads to a lower Gap to Optimum, proving our initial assumption correct at least for this metric.

Table 3.14: Pairwise permutation comparisons of `osfac` levels on metric `gap_to_optimum` (strategy=`ADEI`, n_max=200, osfac=1 000, exrat=0.0, samples_per_it=5).

group1	group2	diff_mean	p_adj_Holm
2	100	0.086	0.013*
2	1 000	0.152	0.000***
100	1 000	0.066	0.002**

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$

As previously observed with other parameters, the improvement in the Gap to Optimum seems to come with a negative effect on the global RMSE, at least in the case of the `ADEI` strategy. This can be seen in the box plot shown in Figure 3.3, where increasing `osfac` leads to a higher global RMSE.

Table 3.15: Pairwise permutation comparisons of `osfac` levels on metric `rmse` (strategy=`ADEI`, n_max=200, osfac=1 000, exrat=0.0, samples_per_it=5).

group1	group2	diff_mean	p_adj_Holm
2	100	-0.028	0.000***
2	1 000	-0.035	0.000***
100	1 000	-0.007	0.260

$^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$

In summary, larger candidate pools improve optimisation performance for `ADEI` by reducing the Gap to Optimum but degrade global model quality as measured by the RMSE, highlighting a clear trade-off between these objectives.

Effect of the Exploitation Ratio `exrat`

Finally, we consider the exploitation ratio `exrat` with ratios of 0.0, 0.05, 0.1, and 0.2 while keeping all other parameters at their default values. Varying `exrat` while keeping all other

Table 3.16: ADEI Summary statistics for metric `rmse` by `osfac` (strategy=`ADEI`).

osfac	mean	median	std
2	0.085	0.084	0.008
100	0.113	0.110	0.019
1 000	0.120	0.117	0.021

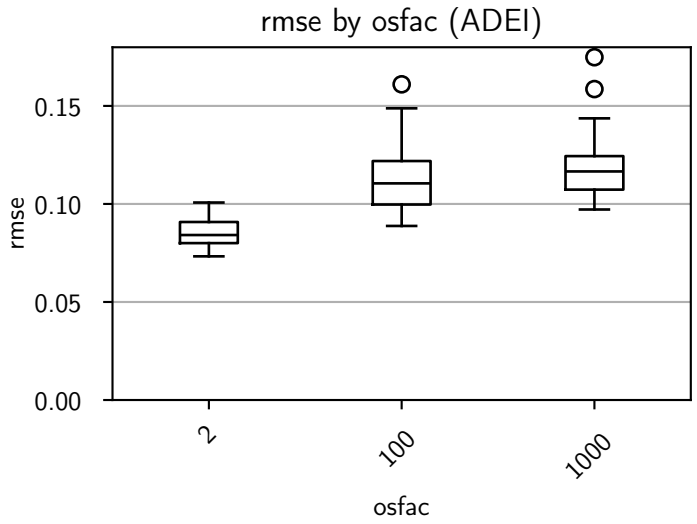


Figure 3.3: RMSE for different `osfac`. The data can be seen in Table 3.16.

parameters at their default values does not yield the results anticipated. The respective one-way permutation ANOVA shown in Table 3.17 indicates statistically significant differences only for the global RMSE. The corresponding summary table Table 3.18 shows that, with increasing `exrat`, the global RMSE tends to increase. However, the Gap to Optimum metric does not change substantially, as indicated by the mean and STD values in Table 3.19. Hence, in this setting a purely exploitative phase at the end, during which optimisation algorithms are run on the SM, does not improve the Gap to Optimum as initially assumed and has a statistically significant negative effect on the global RMSE.

Table 3.17: Global one-way permutation ANOVA for `exrat`

metric	F_perm	p_perm
gap_to_optimum	0.687	0.566
rmse	8.944	0.000***
rmse_near50p	0.511	0.675
rmse_near25p	0.195	0.897
rmse_near10p	0.046	0.986
rmse_near5p	0.013	0.998

$^{*}\,p < 0.05$; $^{**}\,p < 0.01$; $^{***}\,p < 0.001$

Table 3.18: Summary statistics (mean, median, std) for metric `rmse` by `exrat` (strategy=`ADEI`).

exrat	mean	median	std
0.000	0.120	0.117	0.021
0.050	0.130	0.127	0.020
0.100	0.135	0.135	0.012
0.200	0.147	0.146	0.013

Table 3.19: Summary statistics (mean, median, std) for metric `gap_to_optimum` by `exrat` (strategy=`ADEI`).

exrat	mean	median	std
0.000	0.033	0.009	0.036
0.050	0.027	0.001	0.040
0.100	0.020	0.002	0.028
0.200	0.020	0.002	0.029

In summary, for `ADEI` the introduction of a final exploitative phase controlled by `exrat` cannot be recommended in this setting, as it seems to deteriorate global model quality without yielding noticeable gains in optimisation performance.

3.2.3 Summary of the Effect of Strategy Parameters on Strategies

Effects of the parameters can be seen in all strategies. Table 3.20 summarizes which metrics are significantly affected by changes in the strategy parameters for each strategy based on the one-way permutation ANOVA results.

Based on these results it seems like all pool-based adaptive strategies are sensible to changes in all parameters, except `ADUS`, which did not show significant changes in any metric when varying `samples_per_it`. The `Random` baseline and `SemiAD` are both sensible to changes in `n_max` with `SemiAD` also being sensible to changes in `exrat`. This could be explained by the fact that in this case both strategies are using a pool of size `n_max` to select samples, but the number of samples selected via optimisation in the end are last `n_max` $\times$ `exrat` samples are omitted from

Table 3.20: Qualitative summary of the sensitivity of each strategy to changes in the aDoE parameters, based on one-way permutation ANOVAs followed by Holm-corrected pairwise tests versus the default parameter level. Codes indicate which metrics show at least one significant change from the default level (ANOVA $p < 0.05$ and pairwise Holm-adjusted $p < 0.05$): O = Gap to Optimum, G = global RMSE (rmse), L = 5% RMSE (rmse_near5p). '–' indicates no such evidence.

strategy	n_max	n_init	osfac	samples_per_it	exrat
ADEI	G+O+L	G	G+O	G	G
ADUCB	G	G+L	G+O	G	–
ADUS	G+O+L	G+L	G+L	–	O
Random	G+O+L	–	–	–	–
SemiAD	G+O+L	–	–	–	G+O

the training data, hence the models being trained have a different data base in this case as well.

The individual pairwise comparisons for each significant parameter and strategy can be found in the Appendix Section A.3.

4 Test-Rig Experiment

This Chapter applies the aDoE framework to a hardware-in-the-loop road-to-rig test facility, extending the simulation-based study from Chapter 3 to a representative industrial environment. The primary objective is to demonstrate that the proposed approach can be embedded into the existing optimisation workflow and operated robustly under realistic conditions and infrastructure constraints. Beyond this proof of feasibility, the Chapter examines the practical implications of key framework design choices and contrasts the behaviour observed on the test-rig with the simulative results, with a particular focus on aspects relevant for series calibration. The remainder of this Chapter first introduces the experimental setup and methodology and then details the integration of aDoE into the existing tool chain in Section 4.1, before discussing the resulting optimisation performance in Section 4.2.

4.1 Setup and Methodology

The practical implementation of the aDoE experiment on the test-rig was carried out early in the research phase due to limited availability of the test facility. Consequently, the design of the aDoE framework and the selection of sampling strategies could not take into account the insights gained from the simulation-based experiment introduced above in Chapter 3.

The test-rig experiment pursues two main goals. First, it aims to demonstrate that the aDoE framework can be integrated into the existing industrial tool chain and operated robustly in an asynchronous manner. Second, it investigates whether aDoE can achieve optimisation results of comparable or better quality than a static-DoE while using fewer measurements and thereby reducing the overall test-rig time.

For the experiments, an Road-to-Rig (R2R) test-rig from KS Engineers was used in combination with a P1-hybrid, rear-wheel-driven, four-cylinder diesel vehicle. The corresponding vehicle specifications are summarised in Table 4.1. Hybrid vehicles are commonly classified as P0–P4 according to the position of the electric machine in the powertrain. For a P1 mild hybrid, the electric machine is directly connected to the crankshaft. The electric machine, in this context also called an Integrated Starter Generator (ISG), is directly built onto the crankshaft, which is particularly beneficial for comfort-related interventions, e.g. during engine start or for torque assist during acceleration [1]. The increase in degrees of freedom for optimisation that comes with any type of hybrid leads to a greater level of complexity in the manoeuvre under consideration in this work. Further information on hybrid vehicles in general, and the different P configurations in particular, can be found in [1].

The manoeuvre to be optimised in this work is the COM-ES manoeuvre as introduced in Section 2.1.

4.1.1 Test-Rig Integration

Integrating the framework with the KS Engineers R2R test-rig requires coordinating several independent software tools that jointly control the test-bench workflow. Reliable data exchange during runtime is essential and introduces several integration challenges. The test-rig itself mounts the vehicle on four load machines that accurately simulate road load and driving resistance for each tyre individually. This type of test-rig allows realistic simulation and even

Table 4.1: Vehicle specifications

Hybridization	
Degree	Mild-Hybrid
Topology	P1
Combustion Engine	
Fuel	Diesel
Number of cylinders / arrangement	4 / Inline
Displacement	1993 cm^3
Rated power	147 kW
Max. torque	440 N m
Electric Machine	
Design	PMSM
Rated power	17 kW
Max. torque	205 N m
Battery	
Cell technology	Lithium-ion
Rated voltage	48 V
Capacity	17.5 A h
Transmission	
Driven wheels	2
Number of gears	9
Configuration	Automatic with hydrodynamic torque converter and lock-up clutch (LUC)

allows to simulate different kinds of road conditions by actuators which actually apply the resulting road forces to the vehicles suspension. Figure 4.1 shows the test-rig used for the experiments, showing a mounted vehicle and the four load machines. The vehicle itself is an unmodified, fully functional car.

AVL Cameo[1] acts as the central DoE execution environment. It iterates through the test plan, performs conditioning of the vehicle and safety checks, and manages the execution of measurement tasks. ETAS INCA[2] interfaces with the ECUs, providing access to sensor values, actuator signals, and calibration parameters. KS Engineers Tornado[3] controls the physical components of the test-rig and monitors it. AVL DRIVE[4] handles the computation of the scalar target variables from time-series data. MATLAB is invoked by AVL DRIVE to calculate the requested objective from the measurement data.

In the conventional workflow, AVL Cameo, ETAS INCA, and Tornado operate together reliably. For this work, AVL DRIVE is integrated into the stack to process the measurement data ad-hoc and calculate the scalar target values to pass onto the aDoE framework, whereas for static-DoEs

[1] `https://www.avl.com/de-de/testing-solutions/all-testing-products-and-software/connected-development-software-tools/avl-cameo-5` (visited on 2023-10-09)

[2] `https://www.etas.com/ww/en/products-services/data-acquisition-processing-tools/software-products/inca-software-products/` (visited on 2023-10-09)

[3] `https://www.ksengineers.com/Automotive-Testing/Management-und-Automation-Tools/Tornado-Software-Suite` (visited on 2023-10-09)

[4] `https://www.avl.com/de-de/testing-solutions/all-testing-products-and-software/connected-development-software-tools/avl-drive` (visited on 2023-10-09)

Figure 4.1: KS Engineers R2R test-rig with vehicle mounted. Image from [106]

these targets are usually computed on-demand after all measurements have been recorded as they are not required during the experiment. The communication between tools is shown in Figure 4.2.

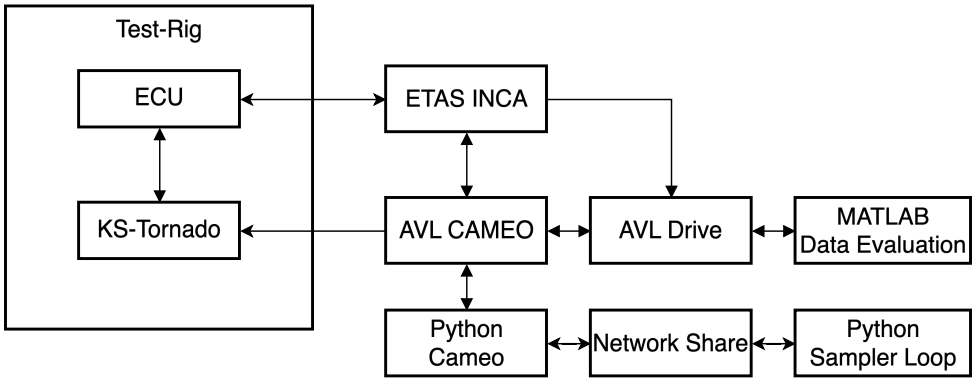


Figure 4.2: Test-rig tool communication.

The aDoE framework is integrated via the AVL Cameo python API. Instead of selecting the next operating point from a predefined test plan, as it is done for static-DoE, Cameo calls a python script, passes the most recent target variables as input, and retrieves the next candidate from the framework, which is then returned to Cameo.

Embedding the full aDoE logic directly into the Cameo script would cause long execution times, block the Cameo interface, and increase idle test-bench time where no manoeuvre is executed. Furthermore, the test-rig workstation is limited to four CPU cores and 8 GB RAM, which is insufficient for fitting complex SM when it is vital to minimize model update times.

For these reasons, the computationally intensive parts of the aDoE pipeline are executed on a separate workstation, leading to the design of an asynchronous experiment and sampling framework. The external workstation continuously produces batches of 20 candidates and

stores them as JSON files on a shared network drive, whereas the Cameo script always uses the latest unused candidate. Early in the experiment, SM updates are fast when using a GPR architecture due to the small dataset and the property of GPR to be very efficient with small datasets. Later, model updates take significantly longer, while the test-bench measurement time per candidate remains the same throughout the experiment with 1 min. This asymmetry is not deliberately desired but can even be beneficial, as early model updates influence the surrogate more strongly in terms of model quality.

A central design aspect is the file-based interface between the test-rig and the external workstation. The sampler writes newly generated candidate sets at each iteration to a new JSON file whose filename contains a timestamp, and the test-rig always reads the latest fully written candidate file. The measurements from the test-rig are appended to a separate JSON file, which the sampler reads when fitting a new model. In this way, parallel write access to the same file is avoided, and the risk of corrupted files due to simultaneous access is mitigated. In the case where the reading and writing is parallel, both scripts throw an error when trying the incomplete JSON. This error is caught and handled by waiting and retrying after a short delay.

Overall, the framework operates in two parallel loops: one on the test-rig workstation, which controls experiment execution and data acquisition, and one on the external workstation, which generates new candidates based on the latest available measurements. The Cameo-side logic is shown in Algorithm 1.

Algorithm 1 Test-rig Cameo Script.

```
1:  while experiment not finished do
2:      Load measurement data from previous iteration
3:      Validate measurement
4:      if measurement invalid then
5:          Compare with past invalid measurements
6:          if operation point repeated then
7:                  Append to blacklist
8:          end if
9:      end if
10:     Append measurement to JSON file on network drive
11:     Load latest candidates JSON file
12:     Pass next unused candidate to Cameo
13: end whileSend termination signal to Cameo
```

The sampling loop running on the external workstation is given in Algorithm 2.

4.1.2 Strategy Setup

The workstation-side candidate generator is modular, enabling the exchange of model architectures and the flexible adjustment of exploration and exploitation strategies for future work. A random candidate generator is used both for constructing the initial design and for sampling new candidates. Instead of generating the complete pool up front, as was done for the simulative experiment, the candidate pool is regenerated in each iteration.

In contrast to the simulation experiments, LHS is not used. The test-rig design space is highly constrained, combining several inequality constraints with blacklisted operating points obtained from earlier measurements. The parameter bounds and constraints are listed in Table 2.1.

Algorithm 2 Workstation sampler loop.

while experiment not finished and new data available **do**
  Load new valid measurements from JSON file
  if number of samples $<$ `n_init` **then**
    Initial design phase
    Generate `n_init` randomly sampled candidates
    Model-based phases
      Fit surrogate model
  else if number of samples $\geq$ `n_max` $\times$ $(1 -$ `exrat`$)$ **then**
      Exploration:
        Evaluate acquisition on 200 $\times$ `n_max` ad-hoc randomly sampled candidates
        Select 20 candidates with highest acquisition values
  else
      Exploitation:
        Run NSGA-II on surrogate model and obtain Pareto front
        Draw 20 random candidates from solutions dominating the reference
    if operation point grid exists **then**
          Choose operation points with highest uncertainty for each solution
    end if
  end if
  Save candidate sets to JSON file
  Update experiment status
end while

Only about 0.08228 % of the unconstrained 11-dimensional space is feasible.[5]

Because the feasible proportion is extremely small, min-max-optimised LHS becomes inefficient due to extensive rejection sampling. Instead, points are sampled uniformly at random from the 11-dimensional hypercube using NumPy and rejected until the requested number of feasible candidates is obtained. For the implementation running on the test-rig, each iteration generates a pool of 20 $\times$ 200 random samples from which the next 20 suggestions are selected. 20 in this case is the candidate batch size which is selected on each sampler iteration and 200 is similar to the `osfac` parameter from the simulative experiment, but applied every iteration instead of only once at the beginning of the exploration phase.

As mentioned in Section 2.1, the COM-ES manoeuvre is defined by 11 parameters, three of which represent environmental parameters and eight of which control the speed gradients of the engine and transmission. Not all eleven parameters serve as optimisation variables. The parameters and their constraints are given in Table 2.1. For AFs that include exploitation, the acquisition value must be evaluated across the complete operation-point space to avoid the model becoming biased towards inherently favourable operation points.

The same principle is already used in the optimisation stage of the static-DoE workflow, where 474 operation points form the basis for computing mean target-function values. The aDoE method generates a pool of random parameter sets, evaluates each set over all operation points for its AF, and sorts the sets by mean AF. The highest-scoring set is then assigned the operation

[5] Estimated via Monte Carlo sampling with 10^7 points and computing the proportion of feasible points.

point that yields the highest predictive uncertainty.

As briefly mentioned in Algorithm 2, a heuristic was added to identify operation points that repeatedly cause invalid measurements, for example when manoeuvres are not executed because the engine shutdown was not initiated or similar. Without such a mechanism, the sampler might repeatedly select these points, as no new data would reduce its uncertainty in that region. This issue occurred mainly in corner regions of the operation-point space for the COM-ES manoeuvre. This problem does not occur for the other eight parameters, as they only take effect once the manoeuvre has already started to be executed. Consequently, they do not lead to invalid measurements. The heuristic runs inside the sampler loop on the workstation. At each iteration, it analyses invalid measurements for repeated patterns of two or three operation points. If such patterns occur more than twice, they are added to a blacklist, which is enforced during rejection sampling.

The developed aDoE framework aims to replace the static-DoE in step 3 of the workflow shown in Figure 2.1 with an adaptive sampling strategy, while keeping all other steps unchanged. In this work, three different sampling strategies were run on the test-rig, each with a budget of 1 000 samples, and compared against a static-DoE baseline with over 2 300 samples in total. The strategies executed on the test-rig differ slightly from those used in the simulative experiment, as the present task is a MOO problem and the AFs must be adapted accordingly. In addition, not all targets are equally easy to learn for a given model, which can lead to high model quality for one target but low quality for another. Therefore, a weighting mechanism is introduced which handles the scalarisation of the different targets AF values as presented below.

Besides using different AFs, each implemented strategy also uses a different method to cope with the multi-objective nature of the problem by scalarizing it using different approaches. The strategies are:

- US Round-Robin (pure explorative).
- R^2-weighted scaled mean-US (pure explorative).
- Scaled mean EI (explorative and exploitative).

In the US Round-Robin strategy, one target is selected in each iteration to choose the next candidates solely based on that target's US. Usually, in Round-Robin approaches, the target is alternated in each iteration. However, due to the asynchronous nature of the test-rig experiment, where measurements are taken continuously and independently of candidate generation, each sampler iteration, yielding 20 candidates, alternates between the targets instead. This means that the test-rig keeps measuring candidates from the same target until the sampler produces a new batch of candidates, which then switches to the other target.

In the R^2-weighted scaled mean-US strategy, each point in the candidate pool is evaluated by taking the US value of each target, dividing it by the mean predicted target value of the candidate pool, and multiplying it by a weight calculated as $1 - R^2$ of the respective target's latest model. The intuition behind this is to prioritise candidates that are expected to improve the target model quality with the lower R^2 value. When the models for both targets are of similar quality, the US values are weighted equally.

In the scaled mean EI strategy, EI is used as the AF. Here, the predicted value and US are combined to form EI as introduced in Subsection 2.3.2, which is then scaled by the respective target's pool mean and aggregated into a single scalar by taking the creating a sum of the scaled

EI values of both targets. As this AF has an exploitative component simply summing the EI values is insufficient as it would favour the operation points with the highest predicted target values. Hence, the EI for each candidate is analysed using the mean EI over the full operation-parameter space analogous to how it is done during the exploitation phase explained in detail below.

In addition, this work experimented with adding an exploitation phase at the end of the experiment, analogous to the `exrat` parameter from the simulative experiment. A value of 0.3 was chosen, meaning that after 700 samples had been selected via exploration, the sampler started to select candidates based on optimising the surrogate model using NSGA-II. As introduced in Subsection 2.1.4, eight parameters must be optimised for two targets while considering the whole operation-parameter space for each candidate. This is a comparably time consuming process, as the full operation-point space must be evaluated for each candidate in the population during each generation. To optimize the solutions over the complete operation-parameter space the same grid of 474 points was used which is used for the nummerical optimisation in the status-quo workflow. The solutions obtained from the optimisation were then filtered for dominating solutions relative to the reference configuration, and 20 random candidates were drawn from these dominating solutions. The dominating solutions are used to reconstruct an 11-dimensional candidate pool from which the next candidates are drawn using US. This leads to solutions being measured at operation points that exhibit the highest uncertainty within the dominating solution space.

4.2 Discussion of Test-Rig Experiment Results

The test-rig experiment was conducted both to demonstrate that aDoE can be technically implemented on the existing infrastructure and to show that initial experiments already yield promising results, even though they can only be validated empirically for now.

ADoE Implementation and Data Quality

From a technical perspective, the implementation of the framework into the existing infrastructure, as described in Subsection 4.1.1, can be considered successful. After iterative improvements and debugging, the system operated reliably throughout the test-rig experiment. The asynchronous architecture, using a network share for data exchange between the test-rig and the sampler running on a remote workstation, proved robust and allowed to measure the maximum amount of samples possible in the time as there was no idle time. Experiments spanning several days could be conducted without human intervention, apart from regular maintenance tasks such as refuelling the car. The fallback to random sampling, in case the data exchange between sampler and test-rig was faulty, was available at all times but was never required during the experiments executed in this work.

The interface between the test-rig and the sampler via the network share also proved to be a reliable means of data exchange with only a few potential failure modes. One key challenge, the possibility of parallel access to the same files by both systems, was addressed by the timestamped-file scheme and the use of a separate measurement file, which avoided parallel write access by design.

An aspect not initially considered at the outset is that aDoE can also help to reduce the number of invalid measurements by analysing invalid measurements online and maintaining a blacklist of operation points that frequently lead to invalid measurements. In a static-DoE experiment

carried out under slightly different vehicle conditions but with a similar scope, more than 2 300 samples were recorded. Of these, 394 measurements were invalid, which can be directly attributed to operation points not leading to the execution of the manoeuvre. These invalid measurements can be accounted for by 71 unique combinations of v_{App} and v_{Brk}, some of which alone led to up to 84 invalid measurements caused by one combination of operation-points. In total, more than 17 % of all measurements were invalid.

Analysing this static-DoE data set revealed that specific combinations of v_{App} and v_{Brk} frequently led to invalid measurements. This can be attributed to inaccuracies in the actuation of the brake and accelerator pedals, which are controlled by a driving robot. Repeating essentially the same experiment on a powertrain test-rig, where the signals of the brake and accelerator pedals are sent directly to the ECU without any actuation robot, leads to more precise control of the manoeuvre and thus more reliable and stable execution of manoeuvres in edge regions of the operation-parameter space.

In contrast, the aDoE experiment, running autonomously with a budget of only 1 000 samples, recorded substantially fewer invalid measurements due to the heuristic implemented to blacklist operation points causing repeated invalid measurements. The aDoE strategy `ADEI` (the EI-based strategy) recorded only 16 invalid measurements, which is only 1.6 % of the total budget, as it blacklisted operation points causing more than two invalid measurements. The `ADUS` strategy, in contrast, resulted in 116 invalid measurements which is 11.6 % of the total budget. The heuristic used for blacklisting was to search for repeating combinations of two out of the three operation parameters. If any combination of two operation parameters occurred in more than two invalid measurements, it was added to a blacklist. The difference in the number of invalid measurements between the two strategies can be explained by the fact that invalid measurements mostly occurred in extreme regions of the operation-point space. Such regions are sampled more frequently by US-based strategies, which explicitly target high uncertainty, whereas `ADEI` selects points based on a balance of predicted improvement and uncertainty and therefore tends to avoid extreme operation points where the expected improvement is limited despite high uncertainty.

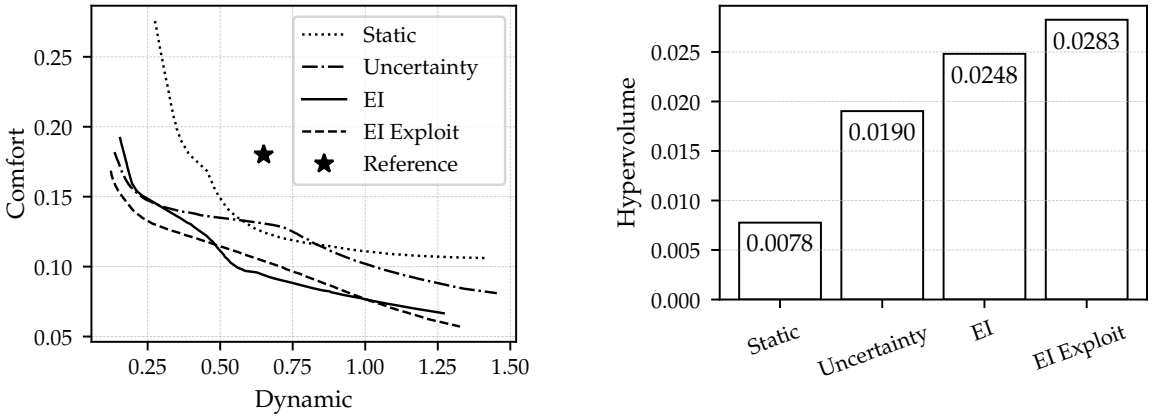
Comparison with Static-DoE and Model-Based Evaluation

The second goal of the test-rig experiment, namely obtaining results that match or surpass the baseline static-DoE, cannot be assessed analytically for several reasons. On the one hand, no full validation with the standard approach was possible within the available test-rig time. Validating a single optimisation solution obtained from the aDoE experiments would take roughly eight hours, and as the PF typically consists of hundreds of solutions, a complete validation quickly becomes infeasible. On the other hand, comparable experiments from previous tasks were carried out under different environmental conditions, leading to differences in the validation data sets and vehicle behaviour. Nevertheless, the solutions obtained from the aDoE experiments could be empirically plausibilised on the test-track and showed promising behaviour, as detailed below.

In total, three different aDoE approaches were executed, each with 1 000 samples. On average, the time from the start of one manoeuvre to the start of the next is approximately 1 min, so that a single experiment comprising 1 000 measurements takes almost 17 hours. To reduce the overall measurement time, the initial test plan with `n_init` = 50 and the validation data set with `n_val` = 50 were executed only once, and for each strategy variation the first 100 measurements were reused. The validation data set is comparatively small, but was chosen due to time constraints

and primarily serves to provide a rough estimate of model quality for live monitoring. As the practical implementation was completed before the simulative experiment, the AFs were chosen based on engineering intuition and the exploratory character of the study and may therefore not represent the best possible choices.

After running the three experiments with three different strategies, the resulting data sets were used to fit GPRs with a more thorough HPO. These models were then used to predict the PF, as introduced in Section 2.1, using the NSGA-II algorithm as implemented in the python package `pymoo`[6] with a population of 300 and 600 generations. Subsequently, the experiment whose predicted PF achieved the highest HV was selected as the most promising candidate for validation on the test-track, with the caveat that this ranking is based solely on model predictions. All three aDoE strategies yielded a higher predicted HV compared to the predicted PF of the static-DoE reference data set, which may indicate an advantage of the adaptive approaches but could equally result from model bias. Consequently, these results cannot be interpreted as conclusive evidence for the superiority of aDoE on the real system.



(a) Pareto front plot of different strategies.

(b) Hypervolume comparison relative to the reference application.

Figure 4.3: Comparison of optimisation results for the test-rig experiment.

Figure 4.3a shows the different PFs obtained from the optimisation based on the models fitted to the respective data sets. Among the three experiments, the run using EI with exploitation predicted the PF with the highest HV. As the fronts in Figure 4.3a and the hypervolume values in Figure 4.3b are obtained purely from model predictions, they must be interpreted with caution. Due to time constraints, only one PF could be evaluated in more detail on the vehicle, so the most promising PF was selected based on the predicted hypervolume. On this basis, the EI-based experiment with the exploitation phase in the end was selected for further investigation and validation on the test-track.

On-Track Validation and Interpretation of Solutions

For the on-track validation, three different solutions from the predicted PF of the `ADEI` experiment were selected. One solution primarily optimises the dynamic criterion, one primarily optimises the comfort-related static criterion, and one represents a compromise between both objectives. The on-track tests confirmed that the three solutions exhibited the expected behaviour relative to each other. The compromise solution was subjectively comparable to the

[6] `https://pymoo.org`

reference, which is the result of a static-DoE.

The parameters identified as solutions by the aDoE approach partly resembled those of the static-DoE solution, shown as the reference in pink. All solutions feature comparatively high values for `Pha_1_0_70`, whereas most solutions from the aDoE strategy suggest lower values for the `Pha_1_90_95` parameter than the reference. Beyond this phase, the trajectories diverge more strongly, with `Pha2_0` ranging from values close to the reference to substantially higher values.

Within the solution set, higher values of `Pha2_0` appear to correlate with lower values of the dynamic criterion, which is desirable as this metric represents a time quantity. Conversely, solutions whose parameter trajectories are more similar to the reference tend to exhibit higher values of the dynamic criterion, which is plausible given their closer resemblance to the existing calibration. Trajectories very similar to the reference solution are coloured in yellow and show a dynamic-criterion value of approximately 0.65, which is also the value of the reference solution.

Overall, the aDoE-derived solutions exhibit a steeper negative gradient in the initial phase of the `Phase 2` parameters, starting at lower values than the reference solution. This behaviour is illustrated in the parallel coordinate plot in Figure 4.4, where the reference solution is highlighted together with the three validation solutions.

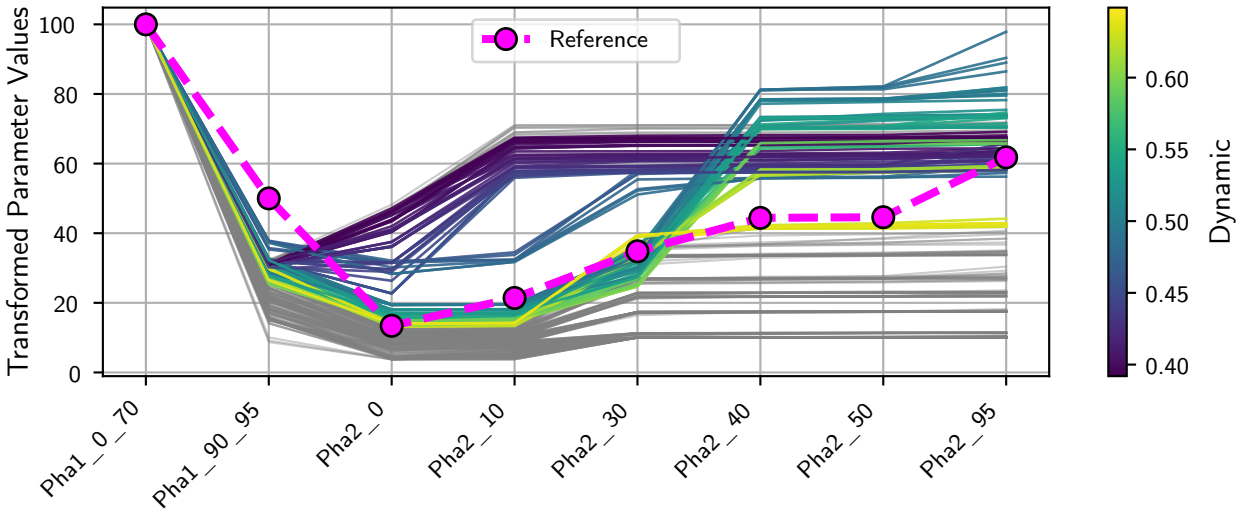


Figure 4.4: Parallel coordinate plot of predicted optimisation solutions for the aDoE with EI. Solutions outperforming the reference are highlighted. An excerpt of the data can be found in Table B.1.

The subjective evaluation indicates that, with an aDoE approach, solutions of quality comparable to the standard approach can be achieved while using less than 60 % of the number of samples required by the standard approach. This roughly translates to a time saving of about 40 %. For a final, comprehensive evaluation, it would be necessary to run every candidate solution across a wide range of operating points, amounting to 474 settings, as is done for the standard approach. Evaluating all 200 solutions on the test-rig would roughly take $474 \cdot 200$ minutes, amounting to almost 66 days of continuous testing. This was not feasible within the scope of this work.

5 Discussion

This chapter synthesises the findings from the simulative experiment in Chapter 3 and the test-rig experiment in Chapter 4 with respect to the research question in Section 1.2. It focuses on the technical feasibility and benefits of aDoE for automotive drivability optimisation and on how the sampling strategies and configuration parameters studied in simulation relate to the behaviour observed on the test-rig.

5.1 Technical Feasibility and Potential Benefits of aDoE

Regarding technical feasibility, the test-rig experiment shows that the aDoE framework can be integrated into the existing drivability-optimisation infrastructure with minor adaptations and without additional hardware or software cost. Communication via a shared network directory and the decoupling of the sampler process from the test-rig environment proved sufficiently robust for multi-day experiments and enabled asynchronous candidate generation and experiment execution. This confirms that the proposed solution is compatible with existing industrial constraints and provides a basis for implementing and comparing further aDoE strategies within the same tool chain.

Beyond enabling aDoE execution on a test-rig, the framework yielded practical benefits not reflected in optimisation metrics. In particular, the implemented blacklisting heuristic contributed to reducing the proportion of invalid manoeuvres from 17 % to below 12 % for the worst-performing aDoE strategy and to below 2 % in the best case by preventing the sampler from repeatedly proposing operation points that systematically fail. The exploitative nature of certain aDoE strategies also contributed to this effect, as they tended to avoid edge regions of the operation-point space where invalid measurements are more likely to occur. Compared with a comparable static-DoE experiment, the adaptive strategies, especially the exploitative `ADEI`-based configuration, thus generated substantially fewer invalid measurements, reducing wasted test-rig time and potential wear in edge regions of the operation-point space.

Regarding potential benefits over static-DoE, the simulative experiment shows that selected adaptive strategies can locate the true optimum with fewer samples than static designs. Specifically, `ADEI` and `ADUCB` achieve a smaller Gap to Optimum than the `Random` and `SemiAD` baselines under the same sample budget and do so more reliably, with a lower STD in Gap to Optimum across repeated runs. The test-rig experiment further indicates that optimisation results of subjectively comparable quality to those from the static-DoE workflow can be obtained with less than 60 % of the measurements, corresponding to a time saving of roughly 40 %, which has to be quantitatively confirmed in future work. Taken together, these findings indicate that aDoE can provide tangible efficiency gains in drivability optimisation.

The results also highlight limitations and trade-offs. All adaptive strategies investigated in the simulative study degraded global model quality compared with the purely space-filling baseline, including nominally explorative configurations such as `ADUS`. This suggests that naively combining exploration and exploitation is insufficient to guarantee both good optimisation performance and high global predictive accuracy. Instead, the current results indicate a fundamental trade-off between improving optimisation performance near the optimum and

maintaining high global model quality, and they imply that aDoE strategies must be explicitly designed to balance these competing objectives.

5.2 Explorative Performance and Sampling-Strategy Behaviour

The simulative experiment shows that purely explorative strategies such as US can perform poorly on the Hartmann-6 benchmark. In terms of global RMSE, `ADUS` does not outperform the static baselines and exhibits the weakest local RMSE in the vicinity of the optimum among all strategies. This is counter-intuitive, as US explicitly targets regions of high predictive uncertainty and might therefore be expected to improve global model quality.

Two main factors can explain this behaviour. First, the GPR architecture used in this work is intentionally rigid in order to ensure fast model updates across many repetitions and does not allow extensive hyperparameter tuning between kernels. As a consequence, the predictive uncertainty is not guaranteed to be well calibrated, particularly in higher-dimensional spaces and sparsely sampled regions. If the predictive variance is overestimated in such regions, US may allocate an excessive share of the sampling budget to areas that are irrelevant for the optimisation objective, improving the model where this is of limited benefit for optimisation. Second, the Hartmann-6 function is a challenging benchmark with multiple local optima and anisotropic structure, which can amplify the tendency of US to concentrate on complicated but suboptimal regions of the design space, as discussed in Section 2.5.

In contrast, the more exploitative strategies `ADEI` and `ADUCB` show markedly different behaviour. Both strategies significantly reduce the Gap to Optimum compared to the baselines, and their local RMSE near the optimum improves more rapidly than that of the explorative strategies. For an aDoE with optimisation as its objective, this is desirable, as it indicates that promising regions can already be identified from a SM trained on comparatively few samples and that sampling effort is focused where it most improves the objective. At the same time, the deterioration in global RMSE for these strategies quantifies the cost of this focus and shows that the resulting models are less suitable for tasks requiring uniformly high predictive accuracy over the full design space. This reduced global model quality might also increase the risk of missing better optima in unexplored regions. A pragmatic implication is that it may be beneficial to start an experiment with a more explorative strategy and switch to a more exploitative one once sufficient global structure has been captured.

The parameter studies around `ADEI` in the simulative setting further clarify how sampling-strategy design controls this balance. Increasing the total budget `n_max` beyond approximately 200–1 000 samples yields only limited additional improvements in the Gap to Optimum metric. Larger candidate pools controlled by `osfac` systematically reduce the Gap to Optimum but increase global RMSE. This is plausible, as larger pools contain more candidates in edge regions which exhibit high uncertainty and therefore attract more samples, degrading global accuracy in regions that are less relevant for the optimum. In contrast, smaller batch sizes `samples_per_it` increase the model-update frequency and, for `ADEI`, improve global RMSE without affecting the Gap to Optimum. The initial design size `n_init` also exhibits a saturation behaviour, where increasing it from 10 to 50 improves global RMSE, while further increases to 100 yield only marginal additional benefits.

These findings also inform the practical configuration of sampling on the test-rig. On the test-rig, `samples_per_it` cannot be controlled directly because of the asynchronous setup and the

objective of keeping the test-rig running continuously, under the assumption that any measurement, even a random candidate or one based on an outdated model, is preferable to idle time in the final data set. At the beginning of the experiment, when only a few samples are available, the GPR update frequency is naturally high, as a single model-fit cycle takes less than a minute, which is consistent with the simulative finding that frequent updates benefit global model quality. Towards the end of the experiment, when model-quality improvements saturate, the lower update frequency becomes less critical.

The exploitation ratio `exrat`, which introduces a purely exploitative phase at the end of the experiment, does not yield the expected performance gains in the simulative setting. For `ADEI`, higher values of `exrat` increase global RMSE while leaving the Gap to Optimum essentially unchanged. Given this data, a final exploitation phase therefore cannot be recommended, as it seems to degrade global model quality without improving optimisation performance. However, for the test-rig experiment, the exploitation phase led to a more optimistic predicted PF, albeit without guarantees of improved predictive accuracy.

Overall, the simulative results indicate that well-performing aDoE strategies in this setting share two properties. First, they rely on acquisition mechanisms that combine exploitation with a controlled amount of exploration rather than on pure US. Second, they employ frequent model updates with moderate candidate pools and avoid strong late-stage exploitation phases. These design principles are now compared with the behaviour observed in the test-rig experiment.

5.3 Transferability from Simulation to Test-Rig and Implications for Drivability Optimisation

The simulative and test-rig experiments differ in several key aspects. The Hartmann-6 benchmark is single-objective, noise-free, and fully known, whereas the COM-ES use case is inherently multi-objective, noisy, constrained, and has an unknown response surface. In simulation, performance can be quantified by global and local RMSE and the Gap to Optimum, while the test-rig experiment cannot be evaluated to the same extent with the available resources. Validating a single optimisation solution for the COM-ES manoeuvre on the test-rig requires roughly eight hours of test-rig time, and a predicted Pareto front typically comprises around 200 or more solutions, so a full quantitative validation would require several months of continuous testing and is therefore infeasible. Consequently, the test-rig aDoE campaign in this work is interpreted primarily as a technical-feasibility demonstration of the online workflow, and the resulting optimisation solutions are regarded as indicative rather than scientifically validated. Candidate solutions from the predicted Pareto front were instead assessed subjectively on track, where they appeared plausible and, in particular, the solution with a similar trade-off as the reference application was perceived to yield comparable drivability. Within these limitations, the simulation and test-rig experiments still show qualitatively consistent trends that inform the design and interpretation of aDoE strategies for drivability optimisation.

First, the test-rig results support the central conclusion from the simulative study that exploitation-heavy acquisition functions are advantageous when optimisation is prioritised over globally accurate modelling. Among the three strategies evaluated on the test-rig, the EI-based configuration with an exploitation phase at the end produced the best predicted Pareto front in terms of hypervolume and generated solution candidates that behaved plausibly on the test-track for different trade-offs between dynamics and comfort. This mirrors the finding that

ADEI and ADUCB outperform the baselines in terms of Gap to Optimum in simulation. Taken together, these results point towards EI-type strategies as promising candidates for drivability calibration and motivate future work to invest resources into at least partial validation of the predicted Pareto front.

Second, the behaviour of purely explorative strategies is consistent across both settings. In simulation, ADUS fails to achieve competitive local model quality near the optimum and offers no advantages over static-DoE baselines in global RMSE. On the test-rig, uncertainty-based strategies are more prone to sampling extreme operation-point combinations, which leads to a higher number of invalid manoeuvres and increased reliance on the blacklisting heuristic. This again suggests that US-type strategies can be problematic when used with a GPR as SM.

Third, the parameter-sensitivity results from the simulative study provide guidance for improving future test-rig campaigns. The findings that moderate budgets, sufficiently large but not excessive initial designs, and frequent model updates benefit ADEI transfer naturally to the COM-ES setting, where model-fitting times and test-rig availability impose similar trade-offs. In the implemented framework, these insights are reflected in the decision to run model fitting on a separate workstation with higher compute capacity and to generate candidates in batches of limited size. Expensive online HPO was deliberately omitted to keep model updates fast, under the assumption that a moderately accurate surrogate model is sufficient to provide reliable guidance for the aDoE process.

Finally, both experiments show that the benefits of aDoE should not be evaluated solely in terms of classical modelling metrics. In drivability calibration, the dominant cost factor is test-bench usage time rather than sample count alone, and the key risks are associated with invalid or unrepresentative manoeuvres rather than purely mechanical wear. The asynchronous architecture and the reduction in invalid measurements observed on the test-rig are therefore important benefits in their own right. Combined with the improved optimisation performance of EI-type strategies, these aspects indicate that aDoE can make drivability-optimisation workflows more time-efficient, robust, and scalable, even if global model quality is not uniformly improved relative to static-DoE.

Overall, the combined evidence from the simulative and test-rig experiments supports an affirmative answer to the research objective formulated in Section 1.2. aDoE approaches can be integrated into existing drivability-optimisation workflows with realistic effort and can provide meaningful benefits in terms of optimisation efficiency and test-rig utilisation, provided that sampling strategies are chosen and configured with care. At the same time, the results emphasise the need for further methodological work to improve the balance between local optimisation performance and global model quality and to better exploit the potential of aDoE in multi-objective, constrained, and noisy automotive applications.

6 Outlook

This chapter discusses potential future work and extensions to the research presented in this thesis. The work primarily focused on analysing different approaches in a simulative environment and demonstrating the feasibility of applying aDoE on a physical test-rig. The simulative study provided insights but also revealed limitations in performance that could be addressed in future research. One of the main challenges observed was the intrinsic tendency of GPR-based uncertainty to favour sampling at the edge of the parameter space when using purely explorative sampling strategies.

Section 6.1 discusses potential improvements to the overall aDoE framework developed in this work, while Section 6.2 outlines possible extensions regarding the sampling strategies and surrogate models used.

6.1 ADoE Framework

Even though the framework developed in this work was functional and robust, there are still many possible improvements required to make it more robust and production-ready. The initial idea was to build a classical server-client infrastructure in which the test-rig would act as a client and the server back-end would run on a workstation. Communication would be based on a network protocol using a standard REST API, and the data would be stored in a database. Such an architecture would allow more efficient data exchange, better oversight, and improved data integrity, as the creation of a large number of temporary files on a network drive could be avoided.

In addition, it would be possible to provide a more universal framework that could be used for any manoeuvre without the need to adapt file paths, network storage locations, and experiment configurations manually on the workstation. Instead, only the client script, which is executed by Cameo, would need to be adapted to specify the experiment by providing the input parameter space, constraints, and objective functions. The server could then run a machine-learning-operations workflow handling several experiments in parallel. All data would be stored in a central database, which would simplify access and live monitoring as well as enable automatic alerting in case of issues. In the longer term, realising the original server-client concept with a dedicated back-end and central database remains an important step towards a production-ready solution.

Other areas of improvement concern the parallelisation of the framework to better utilise the available resources on the workstation. More efficient scheduling and parallel execution of the optimisation loop could enable more extensive HPO, for example by fitting a larger number of models with different architectures in parallel. This, in turn, could improve the quality of the SM and therefore the overall performance of the aDoE process.

The current implementation also does not implement early-stopping mechanisms for the experiments. If no improvement is achieved over several sampler loops, it may be reasonable to stop the experiment early in order to save time and resources and instead dedicate them to validation of the Pareto front or parts of it. Defining robust and interpretable early-stopping rules, for example based on convergence of the estimated Pareto front or stabilisation of model

uncertainty, is therefore an important next step.

Another interesting extension would be to consider not only step 3 of the workflow in Figure 2.1, but to treat steps 2 to 4 in an integrated manner. Instead of starting from a manually pre-selected set of parameters, one could begin with a very generous parameter space including more candidate parameters and automatically assess their importance and select the most relevant ones. The parameter space could then be iteratively reduced to the most influential parameters only, and the actual aDoE experiment would be performed on this reduced space. This would embed automatic feature selection into the experimental-design process and could reduce the required number of experiments while maintaining model quality and optimisation performance.

Additionally, an issue observed during this work was that certain parameter configurations led to failed trials with no successful objective measurements. Because the uncertainty of the surrogate model remains high in such regions, as no new data can be obtained to reduce uncertainty, the sampling algorithm may repeatedly select the same faulty configurations. Instead of implementing a deterministic heuristic that blacklists such configurations after a certain number of failures, as was done in this work, a constraint model could be learned in parallel to the surrogate model in order to predict the feasibility of new configurations. A similar approach has been suggested for another drivability manoeuvre by Prochaska et al. [26], and adapting such constraint modelling to the framework developed here is a promising direction.

From an experimental perspective, a natural next step is to validate a carefully selected subset of Pareto-optimal solutions on the test-rig and test-track. In particular, evaluating extreme and knee-point solutions from the predicted Pareto front for several manoeuvres would allow a more systematic assessment of the external validity and robustness of the aDoE approach.

The implementation developed in this work primarily serves as a proof-of-concept, which was sufficient to evaluate several different strategies and to demonstrate the feasibility of applying aDoE on the test-rig. However, before the framework can be used productively in day-to-day development processes, it needs to become considerably more user-friendly and robust. A graphical user interface that allows users to configure and monitor the experiment and inspect sampled measurements in real time could lower the entry barrier for non-expert users and reduce the need to interact directly with configuration files and logs. Another promising direction is to integrate the aDoE framework more tightly with existing calibration workflows, for example through interactive tools that visualise Pareto fronts, support comparison with reference calibrations, and allow calibration engineers to steer the sampling process or encode preferences. Such human-in-the-loop extensions could help combine the strengths of automated optimisation with domain expertise.

6.2 Sampling Methods

The present work focused on a limited set of SMs and sampling strategies. A natural extension would be to broaden the portfolio of sampling methods supported by the framework, including more advanced SMs architectures. In particular, implementing the HiLoMoT approach [25, 26, 97, 33, 92, 93] would be of high interest, as it may offer a solution to the sampling bias observed towards the edge of the parameter space. Integrating HiLoMoT into the aDoE framework would allow a more flexible approximation of complex response surfaces and could improve sampling efficiency, especially in regions with highly localised phenomena, and could omit

the bias to sample at the edge of the parameter space. Future work should therefore focus on implementing and benchmarking HiLoMoT within the existing framework and on comparing its performance to the SMs and sampling strategies investigated in this thesis.

Beyond HiLoMoT, future work could also explore surrogate models and training schemes with better-calibrated predictive uncertainty, as well as joint modelling of objectives and constraints, in order to make uncertainty-based acquisition functions more reliable in constrained, multi-objective drivability settings. This includes, for example, parallelised online hyperparameter updates, ensemble-based SM, or models explicitly designed to provide reliable uncertainty estimates under limited data and changing operating conditions.

7 Summary

Summary of Findings

This thesis investigated whether aDoE can enhance drivability optimisation within an existing industrial calibration workflow. The proposed framework integrates adaptive sampling into the current test-rig infrastructure using an asynchronous architecture and file-based communication. The test-rig experiments indicate that this integration is technically feasible without additional hardware or major software changes and can be operated robustly over multi-day experiment campaigns.

In a simulative study based on the Hartmann-6 benchmark, several adaptive strategies were compared with static, space-filling baselines. Exploitation-oriented acquisition mechanisms, in particular `ADEI` and `ADUCB`, achieved a consistently smaller Gap to Optimum within the same sample budgets, albeit at the cost of reduced global SM accuracy. The results suggest that, for settings where optimisation is prioritised over uniformly accurate modelling, such strategies can make more efficient use of a limited measurement budget.

A subsequent test-rig experiment applied an EI-based aDoE configuration to the COM-ES manoeuvre. Despite the multi-objective, noisy, and constrained nature of this use case, the adaptive campaign produced plausible Pareto-optimal solutions. Subjective on-track assessments indicated that solutions of quality comparable to those from the established static-DoE workflow can be obtained with less than 60 % of the measurements, corresponding to an estimated time saving of roughly 40 %. Moreover, the combination of blacklisting and more exploitative sampling substantially reduced the fraction of invalid manoeuvres compared with a static-DoE plan. Taken together, these findings constitute an encouraging but tentative indication that aDoE may improve the efficiency and robustness of drivability optimisation, while underscoring the need for further, more systematic validation to quantify its actual impact on drivability outcomes.

Limitations and Open Questions

While the results are encouraging, they come with several limitations that open directions for future work:

- **Trade-off between optimisation and global model quality.** All investigated adaptive strategies degraded global SM quality relative to the space-filling baseline. This indicates a structural tension between improving performance near the optimum and preserving high accuracy over the full design space. More work is needed on strategies that explicitly balance these competing objectives.
- **Limited quantitative validation on the test-rig.** A comprehensive validation of the predicted Pareto front for the COM-ES manoeuvre would require several months of continuous testing and was therefore not feasible. The qualitative conclusions for the real system are thus based on subjective on-track impressions and should be interpreted accordingly. For a definitive assessment, further research has to be conducted.
- **Surrogate-model and uncertainty limitations.** To keep model updates fast, the GPR configuration was deliberately kept simple, with a static kernel configuration and without ex-

tensive hyperparameter optimisation. As a result, the predictive uncertainties might not be well calibrated, which could be a reason for the poor behaviour of purely uncertainty-driven strategies. Exploring richer SMs and improved uncertainty handling remains an open task.

- **Framework maturity and scalability.** The current implementation, based on a shared network directory, was sufficient for prototyping and feasibility studies but is not yet a production-ready solution. A more scalable client-server architecture with central data management could improve transparency, robustness, and reuse across experiments.

Overall, this work represents an initial step towards integrating aDoE into established drivability-optimisation workflows. It demonstrates the potential of aDoE as a useful extension to current practice and identifies concrete directions for advancing strategy design, uncertainty modelling, and validation, thereby supporting its gradual maturation towards broader deployment in drivability optimisation.

Bibliography

Books

[1] Peter Hofmann. *Hybridfahrzeuge. Grundlagen, Komponenten, Fahrzeugbeispiele*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2023. ISBN: 9783662668948 (cit. on pp. 4, 6, 45).

[2] Karl-Ludwig Haken. *Grundlagen der Kraftfahrzeugtechnik. Mit 141 Bildern und 36 Tabellen sowie 20 Übungsaufgaben*. München: Hanser, 2018. ISBN: 9783446455702 (cit. on p. 7).

[3] Alexander I. J. Forrester. *Engineering design via surrogate modelling. A practical guide*. Hoboken, N.J: Wiley, 2008. ISBN: 9780470770801 (cit. on p. 8).

[4] Ping Jiang. *Surrogate model-based engineering design and optimisation*. Singapore: Springer Nature Singapore, 2020. ISBN: 9811507309 (cit. on p. 8).

[5] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Cambridge, Mass.: MIT Press, 2006. ISBN: 026218253X (cit. on pp. 9, 10, 14).

[6] Jack P. C. Kleijnen. 10 Kriging: methods and applications. In: *System- and Data-Driven Methods and Algorithms*. De Gruyter, Oct. 2021, pp. 355–370. ISBN: 9783110498967 (cit. on p. 9).

[7] Douglas C. Montgomery. *Design and analysis of experiments*. Hoboken, NJ, USA: Wiley, 2020. ISBN: 9781119722106 (cit. on pp. 15, 17, 18).

[8] Oliver Nelles. *Nonlinear System Identification*. Springer Berlin Heidelberg, 2001. ISBN: 97 83662043233 (cit. on pp. 15, 16).

[9] Marvin D. Troutt. Regression, 10k Rule of Thumb for. In: *Encyclopedia of Statistical Sciences* ed. by Samuel Kotz et al. 2nd ed., vol. 11. Hoboken, NJ, USA: John Wiley & Sons, Ltd, 2006, p. 7098. ISBN: 9780471667193 (cit. on p. 15).

[10] A C Atkinson, A N Donev and R D Tobias. *Optimum Experimental Designs, with SAS*. Oxford University Press, May 2007. ISBN: 9780199296590 (cit. on pp. 16–18).

[11] Karl Siebertz. *Statistische Versuchsplanung. Design of Experiments (DoE)*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017. ISBN: 9783662557433 (cit. on pp. 17, 18).

[12] Jesús López Fidalgo. *Optimal experimental design. A concise introduction for researchers*. Cham: Springer, 2023. ISBN: 9783031359170 (cit. on p. 18).

[13] Angela M. Dean. *Design and analysis of experiments*. New York: Springer, 2017. ISBN: 978 3319522500 (cit. on p. 18).

[14] Robert L. Mason. *Statistical design and analysis of experiments. With applications to engineering and science*. Hoboken, New Jersey: Wiley-Interscience, 2003. ISBN: 9780471458517 (cit. on p. 18).

[15] Frank Hutter, Holger H. Hoos and Kevin Leyton-Brown. Sequential Model-Based Optimization for General Algorithm Configuration. In: *Learning and Intelligent Optimization*. Springer Berlin Heidelberg, 2011, pp. 507–523. ISBN: 9783642255663 (cit. on p. 19).

[16] Oliver Nelles. *Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes*. Springer International Publishing, 2020. ISBN: 9783030474393 (cit. on p. 19).

[17] Horst Baier, Christoph Seeßelberg and Bernhard Specht. *Optimierung in der Strukturmechanik*. Vieweg+Teubner Verlag, 1994. ISBN: 9783322907004 (cit. on p. 23).

[18] Alexander V. Lotov and Kaisa Miettinen. Visualizing the Pareto Frontier. In: *Multiobjective Optimization*. Springer Berlin Heidelberg, 2008, pp. 213–243. ISBN: 9783540889083 (cit. on p. 23).

[19] Ronald Aylmer Fisher. *The Design of Experiments*. Edinburgh and London: Oliver & Boyd, 1935[BibTeX key: Fisher1935] (cit. on p. 26).

[20] Averill M. Law and W. David Kelton. Simulation modeling and analysis. In: 2. ed. Boston: McGraw-Hill, 2000, p. 583. ISBN: 0071165371 (cit. on p. 32).

Articles, Proceedings and Tech Reports

[21] Luca Castellazzi et al. 'Vehicle Driveability: Dynamic Analysis of Powertrain System Components'. *SAE Technical Paper Series*. 2016. DOI: 10.4271/2016-01-1124 (cit. on pp. 1, 4).

[22] Andreas Riel, Christian Schyr and Eugen Brenner. 'Test driving automotive virtual drivability calibration and BMU'. *Proceedings of the 5th WSEAS International Conference on Simulation, Modelling and Optimization*. 2005. ISBN: 9608457327 (cit. on pp. 1, 4).

[23] Henrik Schmidt, Kay Büttner and Günther Prokop. 'Methods for Virtual Validation of Automotive Powertrain Systems in Terms of Vehicle Drivability—A Systematic Literature Review'. *IEEE Access*. 2023. DOI: 10.1109/access.2023.3257106 (cit. on p. 1).

[24] Josef Zehetner et al. 'Simulation of Driveability in Real-time'. *SAE Technical Paper Series*. 2009. DOI: 10.4271/2009-01-1372 (cit. on p. 6).

[25] Y. Dursun, F. Kirschbaum and Jakobi. 'Approach to adaptive test planning for calibration of longitudinal dynamics of vehicles on test beds'. *6. Internationales Symposium für Entwicklungsmethodik*. 2015[BibTeX key: Dursun2015] (cit. on pp. 6, 19, 22, 28, 29, 60).

[26] Adrian Prochaska et al. 'Approach for Online Design of Experiments with Additional Constraint Modeling'. *Der Antrieb von morgen 2020*. 2021. DOI: 10.1007/978-3-658-35294-3_13 (cit. on pp. 6, 19, 30, 60).

[27] Sebastian Körner, Philipp Skarke and Michael Auerbach. 'Data-Driven Control Unit Calibration for Optimization of Drivability'. *Collaborative Research Advancing Engineering Solutions for Real-World Challenges 3*. 2026. ISBN: 978-3-032-09384-4 (cit. on pp. 6, 30).

[28] Phillip Semler and Martin Weiser. 'Adaptive Gaussian process regression for efficient building of surrogate models in inverse problems'. *Inverse Problems*. 2023. DOI: 10.1088/1361-6420/ad0028 (cit. on p. 6).

[29] Koichi Osawa and Hideya Notani. 'Starting System for Stop/Start with Change of Mind'. *Proceedings of the FISITA 2012 World Automotive Congress*. 2012. DOI: 10.1007/978-3-642-33829-8_21 (cit. on p. 6).

[30] Lukáš Novák et al. 'Active learning-based domain adaptive localized polynomial chaos expansion'. *Mechanical Systems and Signal Processing*. 2023. DOI: 10.1016/j.ymssp.2023.110728 (cit. on p. 9).

[31] Leo Breiman. 'Random Forests'. *Machine Learning*. 2001. DOI: 10.1023/a:1010933404324 (cit. on p. 9).

[32] Kathryn Chaloner and Isabella Verdinelli. 'Bayesian Experimental Design: A Review'. *Statistical Science*. 1995. DOI: 10.1214/ss/1177009939 (cit. on p. 9).

[33] Benjamin Hartmann et al. 'Hierarchical local model trees for design of experiments in the framework of ultrasonic structural health monitoring'. *2011 IEEE International Conference on Control Applications (CCA)*. 2011. DOI: 10.1109/cca.2011.6044489 (cit. on pp. 9, 60).

[34] Georges Matheron. 'Krigeage d'un panneau rectangulaire par sa périphérie'. *Note géostatistique*. 1960[BibTeX key: Matheron1960] (cit. on p. 9).

[35] Donald R. Jones. 'A Taxonomy of Global Optimization Methods Based on Response Surfaces'. *Journal of Global Optimization*. 2001. DOI: 10.1023/a:1012771025575 (cit. on p. 9).

[36] J. Knowles. 'ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimisation problems'. *IEEE Transactions on Evolutionary Computation*. 2006. DOI: 10.1109/tevc.2005.851274 (cit. on pp. 9, 23, 28, 29).

[37] Ciyou Zhu et al. 'Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimisation'. *ACM Transactions on Mathematical Software*. 1997. DOI: 10.1145/279232.279236 (cit. on pp. 11, 23).

[38] Nicholas J. Higham. 'Cholesky factorization'. *WIREs Computational Statistics*. 2009. DOI: 10.1002/wics.18 (cit. on p. 11).

[39] Vladimir Dragalin. 'Adaptive Designs: Terminology and Classification'. *Drug Information Journal*. 2006. DOI: 10.1177/216847900604000408 (cit. on p. 16).

[40] Aleksandar Jankovic, Gaurav Chaudhary and Francesco Goia. 'Designing the design of experiments (DOE) – An investigation on the influence of different factorial designs on the characterization of complex systems'. *Energy and Buildings*. 2021. DOI: 10.1016/j.enbuild.2021.111298 (cit. on p. 16).

[41] James Bergstra and Yoshua Bengio. 'Random search for hyper-parameter optimisation'. *J. Mach. Learn. Res.* 2012. ISSN: 1532-4435 (cit. on p. 17).

[42] Oscar Kempthorne. 'Why randomize?' *Journal of Statistical Planning and Inference*. 1977. DOI: 10.1016/0378-3758(77)90002-7 (cit. on p. 17).

[43] Nicholas Metropolis and S. Ulam. 'The Monte Carlo Method'. *Journal of the American Statistical Association*. 1949. DOI: 10.1080/01621459.1949.10483310 (cit. on p. 17).

[44] Max D. Morris and Toby J. Mitchell. 'Exploratory designs for computational experiments'. *Journal of Statistical Planning and Inference*. 1995. DOI: 10.1016/0378-3758(94)00035-t (cit. on p. 17).

[45] Michael Stein. 'Large Sample Properties of Simulations Using Latin Hypercube Sampling'. *Technometrics*. 1987. DOI: 10.1080/00401706.1987.10488205 (cit. on p. 17).

[46] M. D. Mckay, R. J. Beckman and W. J. Conover. 'A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code'. *Technometrics*. 2000. DOI: 10.1080/00401706.2000.10485979 (cit. on p. 17).

[47] Boxin Tang. 'Orthogonal Array-Based Latin Hypercubes'. *Journal of the American Statistical Association*. 1993. DOI: 10.1080/01621459.1993.10476423 (cit. on p. 17).

[48] Weng Kee Wong. 'G-optimal designs for multi-factor experiments with heteroscedastic errors'. *Journal of Statistical Planning and Inference*. 1994. DOI: 10.1016/0378-3758(94)90146-5 (cit. on p. 17).

[49] J. Kiefer and J. Wolfowitz. 'The Equivalence of Two Extremum Problems'. *Canadian Journal of Mathematics*. 1960. DOI: 10.4153/CJM-1960-030-4 (cit. on p. 17).

[50] M.E. Johnson, L.M. Moore and D. Ylvisaker. 'Minimax and maximin distance designs'. *Journal of Statistical Planning and Inference*. 1990. DOI: 10.1016/0378-3758(90)90122-b (cit. on p. 17).

[51] André I. Khuri and Siuli Mukhopadhyay. 'Response surface methodology'. *WIREs Computational Statistics*. 2010. DOI: 10.1002/wics.73 (cit. on p. 18).

[52] J.C. Helton and F.J. Davis. 'Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems'. *Reliability Engineering & System Safety*. 2003. DOI: 10.1016/s0951-8320(03)00058-9 (cit. on p. 18).

[53] James Bergstra et al. 'Algorithms for Hyper-Parameter Optimization'. *Advances in Neural Information Processing Systems*. 2011. URL: https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf. (Visited on 14/05/2025)[BibTeX key: Bergstra2011] (cit. on p. 19).

[54] Tim Gerling et al. 'Model-Based Sequential Design of Experiments with Machine Learning for Aerospace Systems'. *Aerospace*. 2024. DOI: 10.3390/aerospace11110934 (cit. on p. 19).

[55] Burr Settles. 'Active Learning Literature Survey'. 2009. URL: http://digital.library.wisc.edu/1793/60660. (Visited on 14/05/2025)[BibTeX key: Settles2009] (cit. on p. 19).

[56] David J. C. MacKay. 'Information-Based Objective Functions for Active Data Selection'. *Neural Computation*. 1992. DOI: 10.1162/neco.1992.4.4.590 (cit. on p. 19).

[57] David Cohn, Zoubin Ghahramani and Michael Jordan. 'Active Learning with Statistical Models'. *Advances in Neural Information Processing Systems*. 1994. URL: https://proceedings.neurips.cc/paper_files/paper/1994/file/7f975a56c761db6506eca0b37ce6ec87-Paper.pdf[BibTeX key: Cohn1994] (cit. on p. 19).

[58] Sung Ho Park and Seoung Bum Kim. 'Robust expected model change for active learning in regression'. *Applied Intelligence*. 2019. DOI: 10.1007/s10489-019-01519-z (cit. on pp. 19, 29).

[59] Xiaowei Yue et al. 'Active Learning for Gaussian Process Considering Uncertainties With Application to Shape Control of Composite Fuselage'. *IEEE Transactions on Automation Science and Engineering*. 2021. DOI: 10.1109/tase.2020.2990401 (cit. on p. 19).

[60] Alexandra G. Ellis et al. 'Active learning for efficiently training emulators of computationally expensive mathematical models'. *Statistics in Medicine*. 2020. DOI: 10.1002/sim.8679 (cit. on p. 19).

[61] Bobak Shahriari et al. 'Taking the Human Out of the Loop: A Review of Bayesian Optimization'. *Proceedings of the IEEE*. 2016. DOI: 10.1109/jproc.2015.2494218 (cit. on pp. 20, 23).

[62] Yoshihiko Ozaki et al. 'Multiobjective tree-structured parzen estimator for computationally expensive optimisation problems'. *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 2020. DOI: 10.1145/3377930.3389817 (cit. on p. 20).

[63] Yoshihiko Ozaki et al. 'Multiobjective Tree-Structured Parzen Estimator'. *Journal of Artificial Intelligence Research*. 2022. DOI: 10.1613/jair.1.13188 (cit. on p. 20).

[64] Donald R. Jones, Matthias Schonlau and William J. Welch. 'Efficient Global Optimization of Expensive Black-Box Functions'. *Journal of Global Optimization*. 1998. DOI: 10.1023/a:1008306431147 (cit. on pp. 20, 28, 29).

[65] P. Auer. 'Using upper confidence bounds for online learning'. *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 2000. DOI: 10.1109/sfcs.2000.892116 (cit. on p. 20).

[66] K. Deb et al. 'A fast and elitist multiobjective genetic algorithm: NSGA-II'. *IEEE Transactions on Evolutionary Computation*. 2002. DOI: 10.1109/4235.996017 (cit. on pp. 23, 24).

[67] Kalyanmoy Deb and J. Sundar. 'Reference point based multi-objective optimisation using evolutionary algorithms'. *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. 2006. DOI: 10.1145/1143997.1144112 (cit. on p. 23).

[68] Haitham Seada and Kalyanmoy Deb. 'A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives'. *IEEE Transactions on Evolutionary Computation*. 2016. DOI: 10.1109/tevc.2015.2459718 (cit. on p. 23).

[69] Zitzler, Eckart, Laumanns, Marco and Thiele, Lothar. *SPEA2: Improving the strength pareto evolutionary algorithm*. en. 2001. DOI: 10.3929/ETHZ-A-004284029 (cit. on p. 23).

[70] Xilu Wang et al. 'Recent Advances in Bayesian Optimization'. *ACM Computing Surveys*. 2023. DOI: 10.1145/3582078 (cit. on p. 23).

[71] Stewart Greenhill et al. 'Bayesian Optimization for Adaptive Experimental Design: A Review'. *IEEE Access*. 2020. DOI: 10.1109/access.2020.2966228 (cit. on p. 23).

[72] J. Kennedy and R. Eberhart. 'Particle swarm optimisation'. *Proceedings of ICNN'95 - International Conference on Neural Networks*. 1995. DOI: 10.1109/icnn.1995.488968 (cit. on p. 23).

[73] Rainer Storn and Kenneth Price. 'Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces'. *Journal of Global Optimization*. 1997. DOI: 10.1023/a:1008202821328 (cit. on p. 23).

[74] Sourabh Katoch, Sumit Singh Chauhan and Vijay Kumar. 'A review on genetic algorithm: past, present, and future'. *Multimedia Tools and Applications*. 2020. DOI: 10.1007/s11042-020-10139-6 (cit. on p. 23).

[75] Dong C. Liu and Jorge Nocedal. 'On the limited memory BFGS method for large scale optimisation'. *Mathematical Programming*. 1989. DOI: 10.1007/bf01589116 (cit. on p. 23).

[76] Richard H. Byrd et al. 'A Limited Memory Algorithm for Bound Constrained Optimization'. *SIAM Journal on Scientific Computing*. 1995. DOI: 10.1137/0916069 (cit. on p. 23).

[77] N. Srinivas and Kalyanmoy Deb. 'Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms'. *Evolutionary Computation*. 1994. DOI: 10.1162/evco.1994.2.3.221 (cit. on p. 24).

[78] Anne Auger et al. 'Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point'. *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*. 2009. DOI: 10.1145/1527125.1527138 (cit. on p. 24).

[79] Andreia P. Guerreiro, Carlos M. Fonseca and Luís Paquete. 'The Hypervolume Indicator: Computational Problems and Algorithms'. *ACM Computing Surveys*. 2021. DOI: 10.1145/3453474 (cit. on p. 24).

[80] Victor Picheny, Tobias Wagner and David Ginsbourger. 'A benchmark of kriging-based infill criteria for noisy optimisation'. *Structural and Multidisciplinary Optimization*. 2013. DOI: 10.1007/s00158-013-0919-4 (cit. on pp. 24, 25).

[81] Chen Wang et al. 'An evaluation of adaptive surrogate modeling based optimisation with two benchmark problems'. *Environmental Modelling & Software*. 2014. DOI: 10.1016/j.envsoft.2014.05.026 (cit. on p. 24).

[82] Mingcheng Chen et al. 'ROMO: Retrieval-enhanced Offline Model-based Optimization'. *The Fifth International Conference on Distributed Artificial Intelligence*. 2023. DOI: 10.1145/3627676.3627685 (cit. on p. 24).

[83] S. Huband et al. 'A review of multiobjective test problems and a scalable test problem toolkit'. *IEEE Transactions on Evolutionary Computation*. 2006. DOI: 10.1109/tevc.2005.861417 (cit. on p. 24).

[84] Marti Anderson and Cajo Ter Braak. 'Permutation tests for multi-factorial analysis of variance'. *Journal of Statistical Computation and Simulation*. 2003. DOI: 10.1080/00949650215733 (cit. on p. 26).

[85] Sture Holm. 'A Simple Sequentially Rejective Multiple Test Procedure'. *Scandinavian Journal of Statistics*. 1979. ISSN: 03036898, 14679469 (cit. on p. 27).

[86] Philipp Klein et al. 'Adaptive Test Planning for the Calibration of Combustion Engines – Application'. *Design of Experiments (DoE) in Engine Development*. 2013. URL: https://www.researchgate.net/publication/254560225_Adaptive_Test_Planning_for_the_Calibration_of_Combustion_Engines_--_Application. (Visited on 15/05/2025)[BibTeX key: Klein2013] (cit. on p. 28).

[87] Yehong Zhang et al. 'Near-Optimal Active Learning of Multi-Output Gaussian Processes'. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2016. DOI: 10.1609/aaai.v30i1.10209 (cit. on pp. 28, 29).

[88] Ling Zhu et al. 'Adaptive Design of Experiments for Automotive Engine Applications Using Concurrent Bayesian Optimization'. *ASME Letters in Dynamic Systems and Control*. 2022. DOI: 10.1115/1.4054222 (cit. on p. 29).

[89] Xiuyong Shi et al. 'Application of the Gaussian Process Regression Method Based on a Combined Kernel Function in Engine Performance Prediction'. *ACS Omega*. 2022. DOI: 10.1021/acsomega.2c05952 (cit. on p. 30).

[90] Mohamed Ali Boussaidi et al. 'Random Sampling High Dimensional Model Representation Gaussian Process Regression (RS-HDMR-GPR) for Multivariate Function Representation: Application to Molecular Potential Energy Surfaces'. *The Journal of Physical Chemistry A*. 2020. DOI: 10.1021/acs.jpca.0c05935 (cit. on p. 30).

[91] Jonathan Foldager et al. 'On the role of model uncertainties in Bayesian optimisation'. *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*. 2023. URL: https://proceedings.mlr.press/v216/foldager23a.html[BibTeX key: Foldager2023] (cit. on p. 30).

[92] Mark R. Courtier, Anthony J. Croxford and Kathryn Atherton. 'An iterative design of experiments based data collection approach for ultrasonic guided waves'. *AIP Conference Proceedings*. 2017. DOI: 10.1063/1.4974580 (cit. on p. 60).

[93] Satar Mahdevari and Mohammad Bagher Khodabakhshi. 'A hierarchical local-model tree for predicting roof displacement in longwall tailgates'. *Neural Computing and Applications*. 2021. DOI: 10.1007/s00521-021-06127-y (cit. on p. 60).

Theses

[94] Tino Naumann. 'Wissensbasierte Optimierungsstrategien für elektronische Steuergeräte an Common-Rail-Dieselmotoren'. Dissertation. Technische Universität Berlin, Universitätsbibliothek (Diss.-Stelle). 2002. URL: https://api-depositonce.tu-berlin.de/server/api/core/bitstreams/8b97a969-246c-4994-9b11-d7698761480a/content. (Visited on 25/05/2025)[BibTeX key: Naumann2002] (cit. on pp. 1, 4).

[95] Julien Pillas. 'Modellbasierte Optimierung dynamischer Fahrmanöver mittels Prüfständen'. Dissertation. TU Darmstadt. 2017. URL: http://tuprints.ulb.tu-darmstadt.de/6685/. (Visited on 14/05/2025)[BibTeX key: Pillas2017] (cit. on pp. 4, 6).

[96] Adrian Prochaska. 'Aktive Ausgangsselektion zur modellbasierten Kalibrierung dynamischer Fahrmanöver'. Technische Universität Dresden. 2022. URL: https://nbn-resolving.org/urn:nbn:de:bsz:14-qucosa2-810979. (Visited on 06/10/2025)[BibTeX key: Prochaska2022] (cit. on p. 9).

[97] Nils Tietze. 'Model-based Calibration of Engine Control Units Using Gaussian Process Regression'. Technische Universität Darmstadt. 2015. URL: https://tuprints.ulb.tu-darmstadt.de/id/eprint/4572. (Visited on 14/05/2025)[BibTeX key: Tietze2015] (cit. on pp. 9, 28, 60).

[98] David Kristjanson Duvenaud. 'Automatic Model Construction with Gaussian Processes'. Dissertation. University of Cambridge. 2014. URL: https://www.cs.toronto.edu/~duvenaud/thesis.pdf. (Visited on 23/05/2025)[BibTeX key: Duvenaud2014] (cit. on pp. 11, 13).

[99] Benjamin Hartmann. 'Lokale Modellnetze zur Identifikation und Versuchsplanung nichtlinearer Systeme'. Dissertation. Universität Siegen. 2014. URL: https://dspace.ub.uni-siegen.de/handle/ubsi/786. (Visited on 14/05/2025)[BibTeX key: Hartmann2013] (cit. on p. 28).

Miscellaneous

[100] *ISO 2631-1:1997 — Mechanical vibration and shock — Evaluation of human exposure to whole-body vibration — Part 1: General requirements*. Geneva, 1997[BibTeX key: ISO2631-1_1997]. (Cit. on p. 8).

[101] Sebastian Raschka. *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. 2018. DOI: 10.48550/ARXIV.1811.12808 (cit. on p. 12).

[102] Jason M. Gregory et al. *Taxonomy of A Decision Support System for Adaptive Experimental Design in Field Robotics*. 2022. DOI: 10.48550/ARXIV.2210.08397 (cit. on p. 16).

[103] Adrian Prochaska, Julien Pillas and Bernard Bäker. *Active Output Selection Strategies for Multiple Learning Regression Models*. 2020. DOI: 10.48550/ARXIV.2011.14307 (cit. on pp. 19, 28–30).

[104] Shuhei Watanabe. *Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance*. 2023. DOI: 10.48550/ARXIV.2304.11127 (cit. on p. 20).

[105] James T. Wilson et al. *The reparameterization trick for acquisition functions*. 2017. URL: https://arxiv.org/abs/1712.00424. (Visited on 02/12/2025)[BibTeX key: Wilson2017] (cit. on p. 20).

[106] Mercedes-Benz AG. *Umfassende Motorenoffensive bei Mercedes-Benz: Stärker, sparsamer und sauberer*. 2016. URL: https://presse.mercedes-benz.at/news-umfassende-motorenoffensive-bei-mercedes-benz?id=43875&menueid=10014&l=deutsch. (Visited on 30/11/2025)[BibTeX key: MercedesBenz2016] (cit. on p. 47).

A Appendix Chapter 3

A.1 Normality and Homoscedasticity Tests

Table A.1: Normality (Shapiro–Wilk) and homogeneity (Levene) test results for all metrics including gap_to_optimum.

metric	n_samples	strategy	shapiro_p	normal_ok	levene_p	homogeneity_ok
rmse	200	Random	0.00606	False	0.01164	False
rmse	200	SemiAD	0.00606	False	0.01164	False
rmse	200	ADUS	0.1567	True	0.01164	False
rmse	200	ADEI	0.6285	True	0.01164	False
rmse	200	ADUCB	0.1363	True	0.01164	False
rmse_near05p	200	Random	0.6048	True	0.008126	False
rmse_near05p	200	SemiAD	0.6048	True	0.008126	False
rmse_near05p	200	ADUS	0.5913	True	0.008126	False
rmse_near05p	200	ADEI	9.853e-05	False	0.008126	False
rmse_near05p	200	ADUCB	9.292e-05	False	0.008126	False
gap_to_optimum	200	Random	0.09179	True	0.001843	False
gap_to_optimum	200	SemiAD	0.09179	True	0.001843	False
gap_to_optimum	200	ADUS	0.005386	False	0.001843	False
gap_to_optimum	200	ADEI	5.913e-07	False	0.001843	False
gap_to_optimum	200	ADUCB	6.997e-06	False	0.001843	False

A.2 Simulative aDoE Default configuration data

Table A.2: RMSE metrics at 100 and 200 samples

(a) Global RMSE (mean +/- std)

n_samples strategy	100	200
Random	0.1301 ± 0.0115	0.0950 ± 0.0085
SemiAD	0.1335 ± 0.0162	0.0950 ± 0.0085
ADUS	0.1490 ± 0.0243	0.1237 ± 0.0155
ADEI	0.1470 ± 0.0112	0.1196 ± 0.0181
ADUCB	0.1878 ± 0.0121	0.1608 ± 0.0198

(b) RMSE 5 % near Optimum (mean +/- std)

n_samples strategy	100	200
Random	0.2312 ± 0.0565	0.1669 ± 0.0403
SemiAD	0.2458 ± 0.0638	0.1669 ± 0.0403
ADUS	0.2742 ± 0.0613	0.2990 ± 0.0673
ADEI	0.1763 ± 0.1273	0.1899 ± 0.1342
ADUCB	0.1753 ± 0.1242	0.2002 ± 0.1270

A.3 Simulative aDoE pairwise tests

A.3.1 ADEI

ADEI Variation of n_max

Table A.3: Pairwise permutation comparisons of `n_max` levels on metric `gap_to_optimum` (strategy=`ADEI`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.068	0.296
100	500	0.084	0.189
100	1 000	0.097	0.007**
200	500	0.016	0.269
200	1 000	0.029	0.007**
500	1 000	0.013	0.189

Table A.4: Pairwise permutation comparisons of `n_max` levels on metric `rmse_f1` (strategy=`ADEI`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.044	0.000***
100	500	0.066	0.000***
100	1 000	0.094	0.000***
200	500	0.022	0.000***
200	1 000	0.050	0.000***
500	1 000	0.028	0.000***

Table A.5: Pairwise permutation comparisons of `n_max` levels on metric `rmse_f1_near5p` (strategy=`ADEI`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	-0.006	0.884
100	500	0.056	0.241
100	1 000	0.104	0.003**
200	500	0.062	0.158
200	1 000	0.109	0.000***
500	1 000	0.047	0.115

ADEI Variation of n_init

Table A.6: Pairwise permutation comparisons of `n_init` levels on metric `rmse_f1` (strategy=`ADEI`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
10	50	0.041	0.001**
10	100	0.040	0.003**
50	100	-0.001	0.946

ADEI Variation of osfac

Table A.7: Pairwise permutation comparisons of `osfac` levels on metric `rmse_f1` (strategy=`ADEI`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
2	100	-0.028	0.000***
2	1 000	-0.035	0.000***
100	1 000	-0.007	0.260

Table A.8: Pairwise permutation comparisons of `osfac` levels on metric `gap_to_optimum` (strategy=`ADEI`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
2	100	0.086	0.013*
2	1 000	0.152	0.000***
100	1 000	0.066	0.002**

ADEI Variation of samples_per_it

Table A.9: Pairwise permutation comparisons of `samples_per_it` levels on metric `rmse_f1` (strategy=`ADEI`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
1	5	0.009	0.532
1	10	-0.001	0.941
1	20	-0.014	0.358
5	10	-0.010	0.358
5	20	-0.023	0.004**
10	20	-0.013	0.102

ADEI Variation of exrat

Table A.10: Pairwise permutation comparisons of `exrat` levels on metric `rmse_f1` (strategy=`ADEI`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
0.000	0.050	-0.009	0.288
0.000	0.100	-0.015	0.028*
0.000	0.200	-0.027	0.000***
0.050	0.100	-0.005	0.321
0.050	0.200	-0.018	0.011*
0.100	0.200	-0.012	0.020*

A.3.2 ADUCB

ADUCB Variation of n_max

Table A.11: Pairwise permutation comparisons of `n_max` levels on metric `rmse_f1` (strategy=`ADUCB`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.029	0.000***
100	500	0.020	0.011*
100	1 000	0.011	0.301
200	500	-0.009	0.374
200	1 000	-0.017	0.047*
500	1 000	-0.009	0.374

ADUCB Variation of n_init

Table A.12: Pairwise permutation comparisons of `n_init` levels on metric `rmse_f1` (strategy=`ADUCB`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
10	50	0.052	0.000***
10	100	0.051	0.016*
50	100	-0.001	0.899

Table A.13: Pairwise permutation comparisons of `n_init` levels on metric `rmse_f1_near5p` (strategy=`ADUCB`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
10	50	0.180	0.011*
10	100	0.168	0.068
50	100	-0.012	0.810

ADUCB Variation of osfac

Table A.14: Pairwise permutation comparisons of `osfac` levels on metric `rmse_f1` (strategy=`ADUCB`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
2	100	-0.058	0.000***
2	1 000	-0.074	0.000***
100	1 000	-0.017	0.015*

Table A.15: Pairwise permutation comparisons of `osfac` levels on metric `gap_to_optimum` (strategy=`ADUCB`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
2	100	0.016	0.934
2	1 000	0.164	0.000***
100	1 000	0.149	0.000***

ADUCB Variation of samples_per_it

Table A.16: Pairwise permutation comparisons of `samples_per_it` levels on metric `rmse_f1` (strategy=`ADUCB`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
1	5	0.002	1.000
1	10	-0.004	1.000
1	20	-0.020	0.012*
5	10	-0.006	0.876
5	20	-0.022	0.003**
10	20	-0.016	0.032*

A.3.3 ADUS

ADUS Variation of n_max

Table A.17: Pairwise permutation comparisons of `n_max` levels on metric `gap_to_optimum` (strategy=`ADUS`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.263	0.025*
100	500	0.529	0.000***
100	1 000	0.529	0.000***
200	500	0.266	0.000***
200	1 000	0.266	0.000***
500	1 000	0.000	0.999

Table A.18: Pairwise permutation comparisons of `n_max` levels on metric `rmse_f1` (strategy=`ADUS`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.041	0.000***
100	500	0.074	0.000***
100	1 000	0.106	0.000***
200	500	0.033	0.000***
200	1 000	0.065	0.000***
500	1 000	0.031	0.000***

Table A.19: Pairwise permutation comparisons of `n_max` levels on metric `rmse_f1_near5p` (strategy=`ADUS`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.052	0.164
100	500	0.058	0.164
100	1 000	0.148	0.000***
200	500	0.006	0.791
200	1 000	0.096	0.000***
500	1 000	0.091	0.000***

ADUS Variation of n_init

Table A.20: Pairwise permutation comparisons of `n_init` levels on metric `gap_to_optimum` (strategy=`ADUS`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
10	50	-0.083	1.000
10	100	-0.000	1.000
50	100	0.083	1.000

Table A.21: Pairwise permutation comparisons of `n_init` levels on metric `rmse_f1` (strategy=`ADUS`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
10	50	0.016	0.050
10	100	0.031	0.000***
50	100	0.014	0.050

Table A.22: Pairwise permutation comparisons of `n_init` levels on metric `rmse_f1_near5p` (strategy=`ADUS`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
10	50	0.086	0.002**
10	100	0.124	0.004**
50	100	0.038	0.154

ADUS Variation of osfac

Table A.23: Pairwise permutation comparisons of `osfac` levels on metric `rmse_f1` (strategy=`ADUS`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
2	100	-0.021	0.000***
2	1 000	-0.027	0.000***
100	1 000	-0.006	0.254

Table A.24: Pairwise permutation comparisons of `osfac` levels on metric `rmse_f1_near5p` (strategy=`ADUS`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
2	100	-0.054	0.020*
2	1 000	-0.053	0.011*
100	1 000	0.001	0.956

ADUS Variation of exrat

Table A.25: Pairwise permutation comparisons of `exrat` levels on metric `gap_to_optimum` (strategy=`ADUS`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
0.000	0.050	0.281	0.001**
0.000	0.100	0.378	0.000***
0.000	0.200	0.399	0.000***
0.050	0.100	0.097	0.021*
0.050	0.200	0.119	0.004**
0.100	0.200	0.021	0.250

A.3.4 Random

Random Variation of n_max

Table A.26: Pairwise permutation comparisons of `n_max` levels on metric `gap_to_optimum` (strategy=`Random`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.305	0.000***
100	500	0.409	0.000***
100	1 000	0.454	0.000***
200	500	0.104	0.018*
200	1 000	0.149	0.000***
500	1 000	0.045	0.110

Table A.27: Pairwise permutation comparisons of `n_max` levels on metric `rmse_f1` (strategy=`Random`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.036	0.000***
100	500	0.075	0.000***
100	1 000	0.094	0.000***
200	500	0.038	0.000***
200	1 000	0.057	0.000***
500	1 000	0.019	0.000***

Table A.28: Pairwise permutation comparisons of `n_max` levels on metric `rmse_f1_near5p` (strategy=`Random`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.068	0.000***
100	500	0.144	0.000***
100	1 000	0.182	0.000***
200	500	0.076	0.000***
200	1 000	0.115	0.000***
500	1 000	0.038	0.000***

A.3.5 SemiAD

SemiAD Variation of n_max

Table A.29: Pairwise permutation comparisons of `n_max` levels on metric `gap_to_optimum` (strategy=`SemiAD`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.305	0.000***
100	500	0.409	0.000***
100	1 000	0.454	0.000***
200	500	0.104	0.018*
200	1 000	0.149	0.000***
500	1 000	0.045	0.110

Table A.30: Pairwise permutation comparisons of `n_max` levels on metric `rmse_f1` (strategy=`SemiAD`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.036	0.000***
100	500	0.075	0.000***
100	1 000	0.094	0.000***
200	500	0.038	0.000***
200	1 000	0.057	0.000***
500	1 000	0.019	0.000***

Table A.31: Pairwise permutation comparisons of `n_max` levels on metric `rmse_f1_near5p` (strategy=`SemiAD`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
100	200	0.068	0.000***
100	500	0.144	0.000***
100	1 000	0.182	0.000***
200	500	0.076	0.000***
200	1 000	0.115	0.000***
500	1 000	0.038	0.000***

SemiAD Variation of exrat

Table A.32: Pairwise permutation comparisons of `exrat` levels on metric `gap_to_optimum` (strategy=`SemiAD`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
0.000	0.050	0.145	0.002**
0.000	0.100	0.213	0.000***
0.000	0.200	0.229	0.000***
0.050	0.100	0.068	0.030*
0.050	0.200	0.084	0.014*
0.100	0.200	0.016	0.485

Table A.33: Pairwise permutation comparisons of `exrat` levels on metric `rmse_f1` (strategy=`SemiAD`, baseline config with one-factor variation).

group1	group2	diff_mean	p_adj_Holm
0.000	0.050	-0.004	0.208
0.000	0.100	-0.008	0.016*
0.000	0.200	-0.019	0.000***
0.050	0.100	-0.004	0.208
0.050	0.200	-0.016	0.000***
0.100	0.200	-0.011	0.016*

B Appendix Chapter 4

B.1 Test-Rig Solutions

Table B.1: Data Parallel Coordinate Plot Solutions ADEI with Exploit. Plot is shown in Figure 4.4.

Pha1_0_70	Pha1_90_95	Pha2_0	Pha2_10	Pha2_30	Pha2_40	Pha2_50	Pha2_95	Dynamic	Comfort
99.982	31.151	47.139	67.450	67.991	68.140	68.215	69.122	0.392	0.179
100.000	30.800	46.266	67.222	67.227	67.315	67.427	67.566	0.393	0.178
99.999	30.625	46.215	66.381	66.436	66.445	66.769	67.120	0.395	0.176
99.999	30.492	45.243	64.365	65.219	65.328	65.424	65.830	0.400	0.172
100.000	30.441	41.048	61.399	62.117	62.173	62.348	63.538	0.412	0.165
99.983	31.566	40.437	61.357	61.530	61.933	62.101	62.587	0.415	0.164
99.983	30.982	36.104	59.252	60.620	60.727	61.387	62.101	0.423	0.162
99.999	30.556	37.508	58.935	59.136	59.590	59.679	65.720	0.425	0.161
99.983	30.893	36.104	59.121	59.373	59.994	60.596	60.788	0.427	0.161
99.981	29.712	31.437	58.740	58.985	59.236	59.263	61.390	0.433	0.159
99.985	31.966	31.564	57.813	58.164	58.574	58.736	59.307	0.438	0.158
99.985	31.966	29.617	57.340	57.569	58.052	58.164	60.315	0.442	0.157
99.985	31.966	29.513	57.358	57.386	57.569	57.582	58.164	0.444	0.156
99.985	30.195	28.886	57.373	57.386	57.535	57.569	58.164	0.445	0.156
99.999	31.110	26.331	56.681	57.936	57.942	58.000	59.174	0.448	0.156
99.985	31.907	22.697	57.344	57.384	57.575	57.745	64.369	0.454	0.155
99.996	31.907	22.697	55.913	57.384	57.502	57.575	57.857	0.459	0.154
99.972	35.317	31.527	32.399	58.142	58.566	58.658	59.602	0.468	0.152
99.999	37.838	32.112	32.368	52.634	56.009	57.557	58.325	0.473	0.151
100.000	37.676	30.164	32.092	55.446	55.857	55.900	57.319	0.475	0.151
99.999	37.461	28.307	31.767	51.103	55.982	56.250	59.265	0.481	0.150
100.000	30.905	19.460	19.509	33.117	81.198	82.134	97.871	0.485	0.149
99.998	32.273	19.343	19.544	34.976	80.999	81.771	89.015	0.487	0.149
99.768	30.813	19.509	19.604	33.106	81.198	82.116	90.374	0.488	0.148
99.993	32.812	19.457	19.461	34.418	78.398	78.625	81.053	0.494	0.146
99.998	32.254	19.461	19.469	29.764	78.449	78.645	81.771	0.499	0.144
100.000	28.624	19.469	19.543	29.041	77.190	77.714	78.245	0.507	0.142
99.999	31.667	19.464	19.526	30.281	73.389	73.682	74.252	0.510	0.141
99.974	31.321	18.009	18.061	35.159	72.932	73.755	73.874	0.512	0.141
99.999	31.321	18.009	18.061	31.785	72.932	73.077	73.303	0.516	0.139
99.999	30.057	17.891	18.179	29.873	72.375	74.274	75.463	0.520	0.139
99.997	30.643	17.038	17.391	30.225	72.620	73.151	73.453	0.526	0.137
99.997	30.643	16.592	16.698	30.454	73.151	73.369	73.453	0.531	0.136
100.000	32.245	16.809	16.880	31.539	70.240	70.243	70.545	0.532	0.136
99.981	30.643	17.010	17.038	27.322	73.095	73.184	73.369	0.534	0.135
99.999	27.746	16.781	16.784	31.551	70.243	70.545	71.531	0.536	0.135
99.993	27.495	15.812	15.971	32.307	70.245	70.808	74.684	0.545	0.134
99.995	27.529	15.812	15.971	32.244	70.257	70.640	73.052	0.546	0.134
99.993	29.905	15.767	15.786	29.815	69.900	70.251	70.640	0.551	0.133
99.993	28.233	15.767	15.786	29.903	69.900	69.975	70.640	0.553	0.132
99.997	28.164	15.328	15.612	34.659	64.879	64.987	65.611	0.560	0.132
99.985	28.898	15.324	15.488	31.680	64.260	64.904	65.614	0.566	0.131
99.997	28.164	15.266	15.328	30.576	65.822	66.111	66.416	0.568	0.130
99.997	28.164	15.266	15.328	28.389	65.822	66.111	66.416	0.574	0.130
99.970	27.940	14.882	14.894	28.248	66.311	66.595	67.339	0.581	0.129
99.984	26.398	14.882	14.912	27.065	65.891	66.394	66.780	0.589	0.128
99.999	27.849	14.798	14.804	24.732	65.354	65.678	65.835	0.597	0.127
99.999	25.430	14.804	14.821	24.732	65.354	65.678	66.002	0.602	0.126
99.994	27.022	13.352	14.236	32.984	58.133	58.358	59.436	0.611	0.126
99.777	27.465	13.707	13.752	32.128	57.843	58.397	58.901	0.616	0.125
99.994	27.466	13.353	13.544	33.515	58.127	58.313	59.313	0.619	0.125
99.996	29.835	14.035	14.143	39.354	42.277	42.869	44.183	0.631	0.123
99.999	29.958	14.035	14.194	38.830	42.093	42.277	42.869	0.632	0.123
99.557	29.646	13.944	14.143	39.354	41.297	41.317	41.875	0.636	0.122