SEMI-AUTOMATICALLY OPTIMIZED CALIBRATION OF
INTERNAL COMBUSTION ENGINES

TIMO BURGGRAF, MICHAEL JOSWIG, MARC E. PFETSCH, MANUEL RADONS,
AND STEFAN ULBRICH

Abstract. Modern combustion engines incorporate a number of actu-
ators and sensors that can be used to control and optimize the perfor-
mance and emissions. We describe a semi-automatic method to simul-
taneously measure and calibrate the actuator settings and the resulting
behavior of the engine. The method includes an adaptive process for
refining the measurements, a data cleaning step, and an optimization
procedure. The optimization works in a discretized space and incorpo-
rates the conditions to describe the dependence between the actuators
and the engine behavior as well as emission bounds. We demonstrate
our method on practical examples.

1. Introduction

Due to the wish to save fossil fuels, stringent maximal emission limits and
challenging customer preferences, modern internal combustion engines (ICEs)
become more and more complex. Indeed, research towards the improved
construction of combustion engines is highly relevant for reaching the CO₂
emission targets set by the European Union, cf. [M*16].

This engine development results in an increasing number of actuators and
sensors. There are currently in the order of ten different actuators and
sensors each. Examples for actuators include the amount of injected fuel,
exhaust recirculation control, air valve angle, etc. Sensors measure, e.g.,
the temperature, maximal point of the cylinder, torque, exhaust emission,
etc. The actuators allow to produce a certain torque and revolution fre-
quency, which describe the two main requirements on the engine in usage.
However, several different settings of the actuators can result in the same
torque/revolution frequency combination. Moreover, their dependence is in-
volved and not known exactly a priori, i.e., it has to be measured and approx-
imated. In this paper we deal with the optimal engine calibration problem,
i.e., to efficiently approximate this dependence by few measurements and to
choose optimal actuators settings.

Research by M. Joswig is carried out in the framework of Matheon supported by Ein-
stein Foundation Berlin. Further partial support by Deutsche Forschungsgemeinschaft
(SFB-TRR 109: “Discretization in Geometry and Dynamics” and SFB-TRR 195: “Sym-
ropic Tools in Mathematics and their Application”) is gratefully acknowledged.

The work of T. Burggraf, M. Pfetsch and S. Ulbrich has been supported by Deutsche
Forschungsgemeinschaft within GSC 233 “Computational Engineering”. M. Pfetsch and
S. Ulbrich have also been supported by the SFB 805 “Control of Uncertainty in Load-
Carrying Structures in Mechanical Engineering” and S. Ulbrich by the GSC 1070 “Energy
Science and Engineering”.

1
The engine calibration problem consists of determining a so-called *engine manifold*, which determines for each torque/revolution frequency combination a corresponding setting for actuators. This manifold is usually discretized and the resulting *solution map* is hard-coded into the engine control unit. The settings on the engine manifold have to be chosen in such a way that they are consistent across different torque/frequency combinations, i.e., the engine manifold should be continuous. Moreover, the resulting settings need to obey several restrictions in order to avoid damage of the engine as well as bounds on the emissions produced. The emissions are measured with respect to so-called *driving cycles*. These are certain prescribed changes in the torque/frequency settings over time which should resemble usage in practice. These driving cycles are applied to the engine on a measurement bench and the resulting emissions have to be bounded.

The calibration process described above involves two main steps. First, the dependence between the actuator settings and the output has to be measured. A naive enumeration of a uniform grid of possible actuator settings and interpolation would require exponential time in the number of actuators. Therefore, one needs to design a process to perform the measurements at interesting areas in order to speed up the measurement process. Additionally, the actuator settings are continuously changed without waiting until a steady-state has been reached. This allows to save time, but also introduces measurement errors like hysteresis effects. Second, one should optimally choose the resulting engine manifold and solution map. Thus, based on the measurements, optimal actuator settings obeying the above-mentioned restrictions should be chosen in order to obtain an approximation of an engine manifold.

In this article we propose a way to solve the engine calibration problem, which consists of the following contributions:

- **Adaptive meshing**: The density of measurements is adapted within areas where the measured function is sensitive with respect to its inputs, while keeping the density of measurements coarse where it does not. This leads to a more accurate representation of the engine manifold than with a uniform grid approach with a fraction of the measurements.
- **Data cleaning**: Before optimization, the measured data are cleaned by filtering out redundancies and noise.
- **Discretization and Optimization**: We discretize the space of measurands in a fashion that fits the format of lookup tables as they are stored in the engine control unit. Using discrete optimization techniques, we select among the measured (and cleaned) data such actuator settings that minimize fuel consumption of the engine while its pollutant emissions conform to current regulations. The selected settings are drivable in the sense that the actuator’s variation speed is bounded in order to prevent engine damage.

We would like to stress that the algorithm described in this article is a *tool for* the engineer, not a replacement. While it runs automatically once its parameters have been set, the setting of these parameters, e.g., the determination of the subset of actuators which is to be varied in a given
situation, requires extensive knowledge and experience. In this sense it is a semi-automatic process.

This article is structured as follows. In Section 2 we provide a brief review of the relevant literature. A high-level description of the mathematical problem is given in Section 3. From this, we arrive at the corresponding steps of our process. Section 4 provides the details of our method. In Section 5 we present a practical case-study, and Section 6 contains the experimental results. We close with a few concluding remarks.

2. State of the Art

In this section we describe the state of the art of calibration methods for the optimization of ICEs. Many commercial and research products exist for the measurement and calibration of ICEs. In the following we give a brief overview of the different measurement and optimization approaches and the products that implement them.

2.1. Determining Data Boundaries. Throughout its operation the measurand values of an ICE have to stay within certain boundaries. Some of these boundaries are set in place to avoid destructive situations, e.g., for cylinder pressures or critical device temperatures. Others are induced, e.g., by emission and noise regulations. The set of actuator settings such that neither of the boundary constraints is violated is called the operational space. It can be detected by different techniques.

One classical method is the measurement via star like measurement paths with a defined step width. Each measurement path or ray will be interrupted if one observable violates a destructive or nondestructive limit. The convex hull of all measurement path points then defines the operational space. This technique is typically automatized and part of most commercial tools, like Cameo [GPFL01] and TopExpert [FEZ04].

The MATLAB toolbox LOLIMOT (local linear model tree) contains a further method, developed by TU Darmstadt and Porsche AG. This tool determines the boundary limits iteratively. The measurement starts with a coarse detection of the operational space. In the following iteration steps, the algorithm defines linear measurement paths, which are orthogonal to the facets of the convex hull of the measured boundary points. These measurement paths will be interrupted if hard or soft boundary conditions are violated [SHI00].

Besides the determination of the operational space, other programs directly monitor the boundary conditions. For example, BMW developed the tool mbminimize, which uses regression models for engine limits [Kos03]. IAV created Rapid Measurement, which facilitates faster measurements by employing mathematical and physical models to compensate latency effects [RKNS07].

2.2. Test Run Planning. Generating a uniform grid of measurements is the most obvious method to study the behavior of an unknown system. However, the cost of this approach is exponential in the number of actuators.

The most common technique for test run planning is the Design of Experiments (DoE) approach. The DoE is a test run planning strategy which
employs mathematical modeling to describe the behavior of a system of observables with a minimal number of measurement points. The classical test plans are grouped into central composite, space-filling, and certain designs which are called D-optimal. The latter approach is followed, e.g., by LOLIMOT, which generates local models via linear neuronal networks [Ise10]. The tool mbminimize combines a D-optimal measurement plan with additional interactive features [Kos03].

2.3. Measurement Methods. The measurement methods can be divided into stationary, quasi-stationary and transient/dynamic measurement. The stationary measurement method is most frequently used. Dynamic measurement techniques are still limited to very small application areas, e.g., the dynamic optimization of emission objectives [SHI00].

During a quasi-stationary measurement, the parameters are varied slowly. Therefore the whole physical system follows the dynamic parameters with a bounded contouring error. The data acquisition permanently records the behavior of the whole system. By varying one or more parameters, one observes the impact instantaneously. By inverting the measurement path it is possible to compensate the contouring error [Ise10]. A local model derivation is possible for up to three parameters [Ise10].

DYNMET is a measurement strategy, in order to train dynamic neuronal networks. The system is stimulated with a variation on every input signal. For nonlinear processes one uses APRBS (amplitude modulated pseudo random binary signal) [Ise10].

2.4. Model-based Optimization. One approach to the calibration process is to approximate the engine behavior by a mathematical model and then to optimize this model exactly. However, ICEs are difficult to describe by mathematical or physical models that cover the whole operational space. Especially models for emission predictions are strongly limited. ICEs are systems which follow the Multiple Input–Multiple Output (MIMO) paradigm. Models often do not contain physical correlations, but describe the engine as a black box system [Ise10].

Typically, one uses polynomial functions and least squares techniques in order to approximate a simple model to measured data. Yet strongly nonlinear regions are often not described well enough by these approaches [Mit00].

A further approach are artificial neuronal networks (ANN), which work well for nonlinear regimes. Furthermore, this black-box approach does not need special knowledge about the ICE [Mit00]. Neuronal networks are separated mainly into Multi-Layer-Perceptrons (MLPs) and radial base function (RBF) networks. The principle of LOLIMOT is similar to RBF networks [Ise10].

The toolbox ASCMO uses the superposition of weighted base functions [KKL10]. The weights are derived automatically by a statistical learning approach (SLA). This method is robust and needs only a small number of data points to train the network. The toolbox also includes a calibration suite for the so-derived models.

As a key drawback model-based approaches, be it via neural networks or polynomial approximations, often fail to recognize critical behavior. This
is due to the fact that the models usually rely on smooth curves which are fitted to a given set of data, whereas the processes underlying the combustion are usually nonsmooth. Hence, for engines calibrated via model-based optimization, the actual fuel consumption and emission values may substantially deviate from the predictions. This is one key motivation for the present work.

3. Mathematical Problem Description

The calibration of an internal combustion engine is the procedure to derive an optimal engine manifold. To reduce the complexity of the model and to enhance practical implementation, these manifolds are discretized to obtain solution maps (see Section 3.2 below).

In our setting, knowledge of the engine behavior with respect to variations of its actuator settings is obtained by means of physical experiments on a test bench. In addition to the revolution frequency, typical actuators include the injected fuel quantity, the injection angle, or the valve pressure. The generated torque of the engine is a measurand.

Technically there is a relevant distinction between direct and controlled actuators. Direct actuators such as injected fuel quantity, injection angle, or valve pressure can be set directly on the engine, while controlled actuators are set indirectly. For example, the revolution frequency is a controlled actuator that is regulated via a brake on the engine shaft. However, in our mathematical model this distinction is not relevant. Here all actuator setting are considered as coordinates of points in the admissible domain $U_{ad}$, while measured values such as the engine torque and exhaust emissions yield coordinates of points in the admissible range $Y_{ad}$.

The aforementioned side constraints include limits on pollutant emission as well as physical requirements such as engine temperature limits. They necessitate that not only torque, but several other output values of the engine are measured as well. A realistic engine model features $m \geq 8$ actuators and $n \geq 14$ measurands. In our mathematical model we represent the relation between the setting of $m$ actuators and $n$ sensor values by a function

$$F : \mathbb{R}^m \rightarrow \mathbb{R}^n.$$ 

Throughout we make the fundamental assumption that $F$ is continuous, but not necessarily differentiable everywhere. Further, we will assume the actuator values to be restricted to a box $U_{ad} \subseteq \mathbb{R}^m$, which we call the admissible domain. The noncritical sensor values define another box $Y_{ad} \subseteq \mathbb{R}^n$, the admissible range. The exact definitions of $U_{ad}$ and $Y_{ad}$ will be stated in the subsequent Section 4.1. The feasible space of $F$ is the set

$$F_{fsb} := \{(u, y) \in U_{ad} \times Y_{ad} \mid y = F(u)\} \subseteq \mathbb{R}^m \times \mathbb{R}^n.$$ 

The actuators correspond to the coordinates of $u$; examples are the revolution frequency and the amount of fuel injected. Typical sensor values, i.e., coordinates of $F(u)$, include the torque and the emission of carbon monoxide.

We denote by “freq” the index of the actuator for revolution frequency and by “torq” the index of the torque sensor values. Then we call

$$\text{OP} := \{(u_{\text{freq}}, y_{\text{torq}}) \mid (u, y) \in F_{fsb}\}$$
Figure 1. Schematic of discretized operation field and driving cycle. Resistance times are represented by gray shades. (Also compare Figures 6 and 7.)

the operation field of $F$. The operation field represents the behavior of the engine with respect to revolution frequency and torque. While our methods are more general, we focus on this particular pair of actuator and sensor values in our analysis.

3.1. Continuous Optimization Problem. For the sake of a concise exposition we now describe an idealized continuous optimization problem which has the actual optimization problem that we want to solve as a natural discretization.

*Optimization Space.* The feasible region of our continuous optimization problem is given by all engine manifolds. Each such manifold is given as the image of a map $M : \text{OP} \rightarrow \mathbb{R}^m$ that assigns actuator settings to a given frequency and torque value pair in the operation field. A basic requirement is that these maps are continuous.

Moreover there are vital additional conditions to consider. Varying actuators too fast might damage the engine. Therefore in the (continuous) final solution map the variation speed of every actuator is bounded by constants $\Delta_a$ for $a \in [m]$, where $[m] := \{1, \ldots, m\}$. This yields the *drivability constraint* for a given map $M$ with respect to the actuator $a$:

\[ |M(f, t)_a - M(f', t')_a| \leq \Delta_a \cdot \| (f, t), (f', t') \|, \]

for all $(f, t), (f', t') \in \text{OP}$, where $\| \cdot \|$ is some norm. Thus, the map $M_a : \text{OP} \rightarrow \mathbb{R}$ is Lipschitz continuous with constant $\Delta_a$. We define $\Omega$ as the set of all maps $M$ which are feasible in the sense that they obey the drivability constraint with respect to each actuator.

*Driving Cycle and Emission Constraints.* In accordance with current government regulations and common test cycles, the engine behavior is optimized...
Table 1. Selection of EURO emission constraints for passenger cars with compression ignition engine.

| Em. Unit | E3 (2001) | E4 (2006) | E5a (2011) | E6b (2015) | E6c (2018) |
|----------|-----------|-----------|------------|------------|------------|
| CO mg/km | 2300      | 1000      | 1000       | 1000       | 1000       |
| HC mg/km | 200       | 100       | 100        | 100        | 100        |
| NOx mg/km| 150       | 80        | 60         | 60         | 60         |
| PN mg/km | –         | –         | 5          | 4.5        | 4.5        |
| PM 1/km  | –         | –         | –          | 6·10^{12} | 6·10^{11} |

with respect to pre-defined scenarios, known as driving cycles. In our continuous model a driving cycle is a time-parametrized curve

\[ \gamma_{dc}: [0, 1] \to \text{OP}, \]

whose purpose is to simulate the phases of acceleration and constant speed of real-world driving patterns. For the sake of the simplicity of the exposition here we will assume that a fixed driving cycle is given; cf. Figure 2 for an example. Yet it is also natural to take the combination of several driving cycles into account, and our approach also covers this slightly more general situation.

An important constraint prescribed by regulations is to bound the resulting emissions along the driving cycle. Emission pollutants include carbon monoxide (CO), hydrocarbons (HC), nitrogen oxides (NOx) as well as particulate matter (PM) and number (PN). Denote by \( E \) the index set that corresponds to pollutant emissions and by \( e_p \) the emission limit for pollutant \( p \) over the driving cycle.

Consider an engine manifold map \( M \in \Omega \) and a pollutant \( p \in E \). The frequency and torque values along \( \gamma_{dc} \) produce actuator settings \( M(\gamma_{dc}(t)) \), which result in output values \( F(M(\gamma_{dc}(t))) \), including the pollutant \( p \). The integral of these values must satisfy the emission constraint for all \( p \in E \):

\[
\int_0^1 [F(M(\gamma_{dc}(t)))]_p \, dt \leq e_p. \tag{2}
\]

For instance, Table 1 shows the diesel engine emission constraints for EURO norms 3-6c (E3-E6c).

The optimization formulation. Using the terminology developed so far, the continuous optimization problem is

\[
\min_{M \in \Omega} \int_0^1 [\gamma_{dc}(t)]_{\text{fuel}} \, dt \quad \int_0^1 [F(M(\gamma_{dc}(t)))]_p \, dt \leq e_p \quad \forall p \in E. \tag{3}
\]

The optimization goal is to minimize the total consumed fuel along the given driving cycle \( \gamma_{dc} \).
3.2. Discretization. To obtain a finite-dimensional problem, the model is discretized to yield (characteristic) engine maps. We consider combinations of \(k\) revolution frequencies and \(k\) torque demands and subdivide the operation field \(OP\) of \(F\) into \(k^2\) congruent rectangles \(OP_{ft}\), where \(f, t \in [k]\) denote the rectangle’s frequency and torque coordinate, respectively. Thus, engine maps are given as \(k \times k\)-matrices which for each combination yield the corresponding value of a particular actuator. They are stored permanently in the engine control unit. A solution map consists of a complete set of engine maps, one for each actuator. In this way a solution map yields a discretization of the continuous solution map described in Section 3.1.

Then, in our terminology, the solution map takes as input a frequency and torque pair \((f, t) \in [k] \times [k]\) and yields as output an admissible actuator setting. The latter is a point \(u \in U_{ad}\) such that \((u_{\text{freq}}, y_{\text{torq}})\) lies in the rectangle of the discretized operation field corresponding to the input coordinates \((f, t)\).

Our goal is to obtain a solution map which is optimal with respect to a given objective, while conforming to several constraints. Since typical constraints are continuous but nonlinear and the solution map itself is discrete, this optimization problem belongs to the wide class of mixed-integer/discrete nonlinear optimization problems, which are often difficult to handle.

Our goal can be formulated as follows: Find a finite set \(\hat{F} := \{(u^1, y^1), (u^2, y^2), \ldots, (u^N, y^N)\}\) of \(N\) points in \(F_{\text{fsb}}\), i.e., a set of points \(u^q \in U_{ad}\) such that \(y^q = F(u^q) \in Y_{ad}\), which allows to compute a sufficiently good approximation to an optimal solution \(M\) of the continuous optimization problem (3). The points \((u^q, y^q)\) are obtained via actual measurements, and thus \(\hat{F}\) is called the data set. Elements of \(\hat{F}\) are called data points. The final selection of the data points is the result of a cycle of measurements and optimization steps, which are the discussed in the sections below.

We will assume that none of the points in the data set lies on the boundary of any rectangle \(OP_{ft}\). In this case, the discretization of the operation field into \(k^2\) rectangles partitions \(\hat{F}\) into sets

\[
S_{ft} = \{(u, y) \in \hat{F} \mid (u_{\text{freq}}, y_{\text{torq}}) \in OP_{ft}\}.
\]

We call each set \(S_{ft}\) a stack, and the entire partition \(k\) \(OP := \{S_{ft} \mid f, t \in [k]\}\) is the \(k\)-operation field of \(F\) with respect to \(\hat{F}\). In the solution map each entire rectangle \(OP_{ft}\) will be represented by a single measurement \(d_{ft} \in S_{ft}\).

We will postpone the question how to pick these representatives.

Discrete drivability constraint. Due to the uniform discretization of the \(k\)-operation field, a discrete drivability constraint merely has to bound the difference between actuator settings of representatives of neighboring rectangles \(OP_{ft}\). That is

\[
|d_{ft} - d_{gs}| \leq \Delta_a \quad \text{for all } a \in [m],
\]

and all tuples \((f, t), (g, s)\), where either \(g = f \pm 1\) or \(s = t \pm 1\).
Figure 2. Operational profile of RANDOM driving cycle.

Discrete emission constraint. To each rectangle $OP_{ft}$ we further associate a weight $\omega_{ft}$. This weight is set to zero if the intersection of $OP_{ft}$ with the image of the curve $\gamma_{dc}$ is empty. Otherwise $\omega_{ft}$ is a positive value that reflects the resistance time, i.e., the duration of the curve $\gamma_{dc}$ staying in the rectangle $OP_{ft}$. The function

$$DC: [k] \times [k] \rightarrow \mathbb{R}_{\geq 0}, \ (f, t) \mapsto \omega_{ft}$$

serves as a discrete analogue of the continuous driving cycle $\gamma_{dc}$. Note that the map $DC$ only records which parts of the operating field are met by $\gamma_{dc}$, but it ignores the order in which this happens.

The discrete emission constraint is now given as

$$\sum_{(f,t) \in [k] \times [k]} \omega_{ft} \cdot [d_{ft}]_p \leq \epsilon_p \quad \text{for all } p \in E.$$  

The practical driving cycles, for instance the New European Driving Cycle (NEDC) or Real World Driving Cycle (RANDOM), are given by operational profiles which map a time to a gear-velocity combination; cf. Figure 2. The weights $\omega_{ft}$ can be derived from these profiles. The revolution frequency at a certain time can be calculated directly from the current gear-velocity combination. For the requested engine torque not only the speed but also the acceleration has to be taken into account, along with the car’s mass, roll drag and air flow resistance.

Discrete Optimization Problem. In the discrete version of problem (3) we are given the set $\hat{F}$, subdivided into $k^2$ stacks $S_{ft}$ with corresponding weights $\omega_{ft}$. The optimization objective is now to pick a set of $k^2$ representatives $d_{ft}$ of the stacks $S_{ft}$ that solve

$$\min_d \sum_{(f,t) \in [k] \times [k]} \omega_{ft} \cdot [d_{ft}]_{\text{fuel}},$$

subject to the discrete drivability constraint (4) and the discrete emission constraint (5).

4. Semi-automatic Calibration

The purpose of this section is to explain the algorithmic core of our method. As its input it is given the domain $U_{ad}$, equipped with a grid, and an evaluation oracle for the function $F$. It returns a set $SOL \subset F_{fsb}$ of representatives
of the \( k \)-operation field (recall that its cardinality equals \( k^2 \)). Algorithm 1 is a first rough sketch, which will be detailed in the following. In practice the evaluation oracle for \( F \) is given by an engine mounted on a test-bench.

Algorithm 1: Engine calibration procedure

**Input:** admissible domain \( U_{ad} \), grid \( G \), admissible range \( Y_{ad} \), data set \( \hat{F} \), precision parameter \( k \), evaluation oracle \( F \)

**Output:** solution map \( SOL \), updated data set \( \hat{F} \)

1. initialization
2. **while** solution map \( SOL \ in incomplete do**
3. iteration step: adds to \( \hat{F} \) and updates \( G \)
4. data cleaning: reduces \( \hat{F} \)
5. select SOL from \( \hat{F} \) via integer linear program
6. **end**
7. return \((SOL, \hat{F})\)

4.1. Initialization. The calibration procedure is initialized with the domain \( U_{ad} \) equipped with a grid \( G \), the set of noncritical target values \( Y_{ad} \), a (possibly empty) set of data points \( \hat{F} \) and a precision parameter \( k \). The function \( F \) is given implicitly; for a given setting of the actuators, the sensors yield the respective function value by means of a physical measurement. We assume that the data points in \( \hat{F} \) reflect true values of \( F \).

The domain

\[
U_{ad} = \prod_{i=1}^{m} I_i
\]

is a product of intervals \( I_1, I_2, \ldots, I_m \), where \( I_i \) defines the range of variation of actuator \( i \). It may happen that \( I_i \) degenerates to a single point. Then the corresponding actuator is called *static*, otherwise it is called *dynamic*.

There are two cases to distinguish. In the first case, the set \( \hat{F} \) of data points is empty. Then, for each dynamic actuator \( i \), the corresponding interval \( I_i \) is subdivided into parts of equal lengths. Otherwise, if \( \hat{F} \) is not empty, then we assume that each interval \( I_i \) is equipped with its own subdivision. In either case, the product form (7) induces a grid structure \( G \) on the domain. If \( \hat{F} \) is empty then this grid \( G \) is uniform. In the later stages of the optimization, however, \( G \) will become more and more non-uniform.

Each measurand has an interval \( J_j \) of noncritical values. For example, the temperature of the test engine has to stay within certain bounds to prevent it from damage. Then

\[
Y_{ad} = \prod_{j=1}^{n} J_j.
\]

In practice, \( Y_{ad} \) is given in part by the physical test engine (e.g., the aforementioned temperature limits) as well as by external factors such as government regulations (e.g., emission limits). Just like \( F \), which is given by the physical test engine, we will assume the set \( Y_{ad} \) to stay fixed throughout
the whole calibration process. The precision parameter $k$ is dictated by the engine control unit’s engine map format.

During the various steps of the calibration, data points will be added to the set $\hat{F}$. Thus, it may happen that $\hat{F}$ becomes prohibitively large to perform subsequent steps of the calibration. How to weed out less relevant measurements is the subject of Section 4.3 below.

### 4.2. Iteration Step. The basic iteration step can be subdivided into two phases: The generation of a measurement plan, followed by the actual measurement, which is combined with a refinement of the grid.

#### Generation of the Measurement Plan. For our given grid $G$ and data set $\hat{F}$, we construct an abstract graph $G = \bar{G}(G, E)$ as follows. The nodes of $G$ are the grid boxes determined by $G$. Two $m$-dimensional grid boxes are joined by an edge if their intersection is a grid box of dimension $m - 1$. If there are no measurements yet, i.e., if $\hat{F}$ is empty, then the grid $G$ is uniform, and the graph $G$ is the dual graph of a cubical cell complex. Due to non-uniform refinement, the structure of $G$ will become more complicated.

The graph $G$ is equipped with nonnegative node and edge weights. For a grid box $B$, we denote by $\#B$ the number of points $(u^q, y^q) \in \hat{F}$ such that $u^q \in B$. Then the weight of a grid box $B$ is chosen as

\[
    w(B) = \frac{\text{vol}(B)}{\#B + 1}.
\]

The quotient $\#B/\text{vol}(B)$ is the data density of $B$. Similarly, we define the weight of an edge between two adjacent grid boxes $B$ and $B'$ as

\[
    w(B, B') = \frac{\text{vol}(B) + \text{vol}(B')}{{\#(B \cup B')} + 1}.
\]

There is some room for adjusting these weights; the general idea is that the weights should be roughly reciprocal to the data density. But they should stay finite even if the data density is zero.

Due to the involved nature of the measurement process, it is not advisable to take arbitrary independent measurements. In fact, the following concept is crucial as it reflects how the actual measurement process is organized on an engine test bench in practice.

**Definition 4.1. (Measurement Ramp)** For two points $u^q, u^r \in U_{ad}$, we call the set

\[
    \left\{ u^q + i \cdot \frac{u^r - u^q}{\ell - 1} \mid i = 0, 1, \ldots, \ell - 1 \right\}
\]

the measurement ramp from $u^q$ to $u^r$ with $\ell$ measurements.

Zero-entries of $u^r - u^q$ correspond to actuators which are locally static. Suppose that there is an admissible point $u = u^M \in U_{ad}$ where the last measurement took place. If this does not exist, we choose $u$ uniformly at random in the domain $U_{ad}$. Let $B$ be the grid box containing $u$. We may assume that $B$ is unique, since $u$ has been constructed in a randomized fashion. The two steps of the generation of the measurement plan are as follows:
Figure 3. The routing during measurement planning prefers regions of low data density. On the right: Corresponding path in the induced abstract graph.

I. Random grid box: Pick a grid box $B'$ at random with probability $w(B')$, where $\overline{w}$ is the sum of the weights of all boxes.

II. Measurement path: Determine a shortest path $B_0 = B, B_1, \ldots, B_s = B'$ in $\mathcal{G}$ from $B$ to $B'$ using Dijkstra’s algorithm with respect to the weights; cf. [Coo98, §2.2]. In each box $B_q$ for $q \in [s]$, pick a point $u^{M+q}$ uniformly at random. Connect the points $u^M, u^{M+1}, \ldots, u^{M+s}$ by $s$ measurement ramps with $\ell$ measurements each.

This will result in a total of up to $s \cdot \ell$ new measurements to be added to the set $\hat{F}$.

Measurement and symmetric grid refinement. The actual measurement is combined with an adaptive refinement of the grid. The goal of the iteration is to fill the solution map in a way that the remaining small holes can be closed by extrapolation of the surrounding data. The next step is then:

III. Measurement: The actuator settings are varied continuously along the path prescribed by the measurement ramps. This results in a linear ordering of the measurements.

For the recording of the $\ell$ measurements per ramp, several additional aspects have to be taken into account, as explained below. It is important not to store every measurement, since we do not want to store redundant data. Thus, we restrict our attention to relevant value variations, i.e., we only store the measurement $F(u^q)$ at a point $u^q$ if the measurement is sufficiently different. Consequently, fewer than $s \cdot \ell$ data points may be added to the set $\hat{F}$. Here one can choose among various meaningful distance functions. However, we also want to establish a minimal measurement frequency. That is, if we omitted too many subsequent measurements due to the previous rule, then we store the measurement nonetheless.

Moreover, it may happen that a measured value lies outside the noncritical range $Y_{ad}$, i.e., it violates one or more restrictions. In order to exclude critical values, the entire measurement is disrupted, and we continue from scratch at (I) with the last valid measurement. Special care is needed for observables with a pronounced latency. The development of these values
is extrapolated, and the measurement is rerouted already if the measured values get sufficiently close to the boundary of \( Y_{ad} \).

4.3. Data Cleaning. Assume that the measurement phase of the iteration step is complete, i.e., a new set

\[
\hat{\mathcal{F}} = \{(u^1, y^1), \ldots, (u^{M'}, y^{M'})\}
\]

is available. In Section 4.4, we will compute local polynomial fits to the function \( F \) using data points in \( \hat{\mathcal{F}} \) as interpolation points. At points in which the interpolation is not good enough, the grid is refined further. A fundamental necessity for the good fit of a polynomial approximation is that the interpolation points from which it is generated resemble a random point cloud. This is not the case for \( \hat{\mathcal{F}} \), since all points lie on a piecewise linear path given by the sequence of measurement ramps. We thus need to extract a subset \( \hat{\mathcal{F}}_{\text{red}} \) from \( \hat{\mathcal{F}} \) that is sufficiently generic.

Furthermore, we want to translate the discrete version (6) of our optimization problem into an integer linear program which picks the right kind of measurements from which we can then obtain our engine maps; cf. Section 4.6 below. However, our measurements need to be preprocessed in order to make such an approach feasible by significantly reducing the amount of data without affecting the accuracy.

As it turns out we can achieve both goals with a single method, which we call the adaptive space compressor. The idea is the following. For the set \( \hat{\mathcal{U}} = \{u^1, u^2, \ldots, u^M\} \), which is the projection of the set \( \hat{\mathcal{F}} \) onto the admissible domain \( U_{ad} \), we define a corresponding threshold graph. Its nodes are the points \( u^q \), and there is an edge between \( u^q \) and \( u^r \) if the Euclidean distance \( \|u^q - u^r\|_2 \) is below a certain threshold. This threshold depends on the sizes of the two grid boxes which contain \( u^q \) and \( u^r \). Then we iteratively remove measurements which have the maximal node degree in the threshold graph until only isolated nodes remain. This reduced set is denoted as \( \hat{\mathcal{U}}_{\text{red}} \), which yields

\[
\hat{F}_{\text{red}} = \{(u, y) \in \hat{\mathcal{F}} \mid u \in \hat{\mathcal{U}}_{\text{red}}\}.
\]

The set \( \hat{F}_{\text{red}} \) has significantly fewer data points compared to \( \hat{\mathcal{F}} \). But due to the criteria for grid refinement, which will be introduced in (V) and (VI) below, the distribution of the remaining data points reflects the structure of the grid \( G \), which in turn reflects the behavior of \( F \). This especially means that the data density is higher in regions where the behavior of \( F \) is hard to reconstruct by interpolations. Thus, the adaptive space compressor filters out the relevant information gained by the preceding measurements. Moreover, it breaks up the piecewise linear structure of the data point distribution, leading to local (pseudo-)randomness which is required by the polynomial interpolation.

Remark 4.1. In our experiments we further observed that this approach has the additional benefit of producing solutions which are stable with respect to hysteresis effects.

Note that this reduction does not affect the set \( \hat{\mathcal{F}} \), but we rather explicitly keep \( \hat{F}_{\text{red}} \) as a subset. In this way, all our measurements are taken into
account for computing the weights according to (8) and (9) in (I) in the next round of measurements.

4.4. Grid Refinement. The purpose of refining the grid $G$ is to accumulate sufficiently many “meaningful” data points in the set $\hat{F}_{\text{red}}$, such that a solution map of $k^2$ data points can be extracted from it which conforms to both emission and drivability constraints. Recall that in our application we have $m \geq 8$ actuators and $n \geq 14$ measurands. This makes computing on a uniform grid infeasible. The price to pay is that it is slightly more involved to determine points $u^q$ in the admissible domain $U_{\text{ad}}$ such that the data point $(u^q, y^q)$ adds significant information to our data set $\hat{F}_{\text{red}}$ and is thus stored.

We proceed with the following step:

IV. Computation of local fit: We employ a Newton interpolation approach, which requires only a partial recomputation of the interpolation polynomial if some interpolation points are exchanged. Each component function $[F]_i : \mathbb{R}^m \rightarrow \mathbb{R}$, where $i \in [n]$, has to be interpolated by a Newton polynomial $L_d[F]_i$, where $d$ is the approximation order. Assuming the $(d + 1)$-times continuous differentiability of $[F]_i$ at $u$, a $d$-th order polynomial fit requires $N := \binom{m+d}{m}$ interpolation points \{u^{q_1}, u^{q_2}, \ldots, u^{q_N}\} \subseteq \hat{U}$. These $N$ points have to satisfy certain spatial relations for the approximation properties of the interpolation polynomials to hold; the interpolation problem is then called poised. The formulation of the latter conditions is somewhat technical. We thus omit the details and refer to the literature instead; cf., e.g., [SHI00]. Our data cleaning method in Section 4.3 takes care of poisedness.

If an interpolation polynomial with $N$ interpolation points which have the required geometrical structure fails to give a $d$-th order approximation of $[F]_i$ at $u$, this indicates insufficient smoothness of $[F]_i$ at $u$, which we will use as a criterion for a local grid refinement. We aim at a second order fit, i.e., $d = 2$, which requires

$$N = \binom{m+2}{m} = \frac{m(m+1)}{2}$$

interpolation points. Our goal is to compare the measured value $y^q \in Y_{\text{ad}}$ at a point $u^q \in U_{\text{ad}}$, where $(u^q, y^q) \in \hat{F}_{\text{red}}$, with a polynomial interpolation of $F$ in $u^q$ using elements of $\hat{F}_{\text{red}}$ as interpolation points. The error of such an approximation is minimized if we use the $N$ closest neighbors \{u^{q_1}, u^{q_2}, \ldots, u^{q_N}\} of u^q in U_{\text{ad}} as interpolation points, such that

$$\{(u^{q_1}, y^{q_1}), (u^{q_2}, y^{q_2}), \ldots, (u^{q_N}, y^{q_N})\} \subseteq \hat{F}_{\text{red}}.$$

Concerning the implementation of the polynomial interpolation: As can be seen in the experimental section below, we will usually have less than 50,000 points in $\hat{F}_{\text{red}}$. This makes the following brute-force approach feasible. Let $M_{\text{red}} := |\hat{F}_{\text{red}}|$ and assume for simplicity that $q = 1$. Now calculate the squared Euclidean distances of $u^2, u^3, \ldots, u^{M_{\text{red}}}$ to $u^1$. This has a cost of roughly $3 \cdot m \cdot M_{\text{red}}$ elementary arithmetic operations. Store these values in the first row of a $(2 \times [M_{\text{red}} - 1])$-array and the indices of the corresponding points $u^r$ in the second row ($\approx 2 \cdot M_{\text{red}}$ writes). Then determine the $N$
smallest entries of the first row and return the corresponding second row entries.

A simple in-place algorithm to accomplish this is the following: First, determine the $N$-th smallest element of the first row, which costs one sweep of the array; cf [CLRS09, p.183 ff.]. Second, sort all columns to the front of the array whose first row entry is smaller than or equal to the $N$-th smallest first row entry. This costs another sweep of the array. Finally, return the first $N$ second row entries.

Repeating this procedure for all $M_{\text{red}}$ points in $\hat{F}_{\text{red}}$, we arrive at an approximate cost of $(3 \cdot m + 4) \cdot M_{\text{req}}^2 \in \mathcal{O}(m \cdot M_{\text{red}}^2)$ elementary operations.

In our setting, for 50,000 points and 8 dynamic actuators, this results in a total of about 70 billion elementary operations, which is a fairly insignificant task for modern computers. For a general discussion on the nearest neighbor search in high dimensions we refer to the survey [AI18].

The next step is the following:

**V. Symmetric grid refinement:** To decide whether the grid needs to be refined, we compare the interpolated value, say $\tilde{F}(u^q)$, with $y_q = F(u^q)$.

If the deviation exceeds a threshold, then the box $B$ which contains $u^q$ is split into $2^m$ smaller grid boxes, symmetrically in all coordinate directions.

The quality of the local fit depends on the differentiability properties of the approximated function. Hence, the symmetric grid refinement increases the measurement density in areas of potential nondifferentiability.

In practice, the static actuators can be ignored for the computation of the local fit and the symmetric grid refinement.

**Making the Grid Nonuniform.** Recall that ultimately we want to solve the discrete version (6) of the constrained optimization problem (3), i.e., we want to minimize the fuel consumption subject to the drivability and emission constraints. For the iteration step, the drivability constraints are irrelevant. Therefore we focus on the subset $E$ of measurands corresponding to the emissions. The final output of the calibration will be a set SOL of $k^2$ data points $d_{ft}$, where $f, t \in [k]$, which cover the $k$-operation field, while satisfying the emission constraints (5). In practice, it is a major challenge to find sufficiently many points which satisfy the conditions (5) imposed by emission control. This leads us to a second type of grid refinement.

**VI. Asymmetric grid refinement:** Consider the point $u^M \in U_{\text{adj}}$ at which the last measurement was performed. Let $f$ and $t$ be such that $(u^M_{\text{freq}}, y^M_{\text{torq}}) \in \text{OP}_{ft}$ for $y^M = F(u^M)$. If, for any $p \in E$, we have

$$[F(u^M)]_p < \min_{(u^q, y^q) \in S_{ft}} ([F(u^q)]_p),$$

i.e., if the emission measurement $[F(u^M)]_p$ is lower than any other value on the corresponding stack $S_{ft} \subset \hat{F}$, a cross-measurement is performed. To this end, each actuator is varied individually to determine the direction with the biggest impact on $[F]_p$. Afterwards, the grid box $B$ containing $u^M$ is split into two congruent (sub-)boxes along the axis.
corresponding to the actuator whose variation has the biggest impact on $[F]_p$.

Both types of grid refinement do not add data points to $\hat{F}_{\text{red}}$ directly. However, the grid refinements increase the probability for picking one of the subboxes in (I). This way one can hope to find data points near $(u^q, y^q)$ that add relevant information about $F$ and near $(u^M, y^M)$ with better emission values than those currently stored in $\hat{F}$.

4.5. Filling the Gaps. The calibration algorithm traces the behavior of the map $F$, given by the physical test engine, by constructing a sequence of (one-dimensional) measurement ramps through the domain $U_{\text{ad}}$ which is high-dimensional, as is the range of $F$. Naturally, this approach cannot produce a sufficient coverage of $U_{\text{ad}}$. In particular, often several stacks $S_{ft}$ of the $k$-operation field will remain empty. If an empty stack $S_{ft}$ lies in the interior of the operation field, we pick $N = m(m + 1)/2$ data points \{(\(u^n, y^n\)), (\(u^{n2}, y^{n2}\)), \ldots, (\(u^{nN}, y^{nN}\))\} from neighboring stacks and close the hole by a Newton type interpolation on \{\(u^{q1}, u^{q2}, \ldots, u^{qN}\)\} as done in the local fit step IV. However, if an empty stack $S_{ft}$ lies on the boundary of the operation field, then the holes are closed by calculating secants of neighboring data points in the nearest stacks and extending them linearly.

4.6. An Integer Linear Program. In this section we detail how to employ an integer linear program to derive a solution $SOL$ from the reduced set of measurements $\hat{F}_{\text{red}}$ that satisfies the objectives listed in Section 3.2. For an introduction to the solution of integer linear programs, see [Coo98], [Sch98], and [Sie01].

Now, for a data point $d^f$, let $S^q$ be the stack that contains it. As before, let “fuel” be the data point index corresponding to the injected fuel quantity. Then the prey value of $d^f$ is defined as

$$p^q := \frac{1}{[d^f]_{\text{fuel}}} \min\{[d^f]_{\text{fuel}} \mid d^f \in S^q\}.$$

For each index $q \in [M]$, where $M := |\hat{F}_{\text{red}}|$, we introduce a binary decision variable $s^q \in \{0, 1\}$, which indicates whether the data point $d^f$ is part of the solution $SOL$. Then the objective function of our integer linear program is defined as

$$\text{(10) minimize } \sum_{q=1}^{M} p^q s^q.$$

The solution $SOL$ should contain at most one element from each stack $S_{ft}$. This condition is reflected in the stack constraint

$$s_{ft} + \sum_{d^f \in S_{ft}} s^q = 1 \text{ for all } (f, t) \in [k] \times [k],$$

where $s_{ft}$ is a stack decision variable; it is 1 if it is not possible to choose a data point for stack $S_{ft}$ and 0 otherwise. The $s_{ft}$ variables will be used in (12) below to assign penalty values to stacks whose data points are not part of the solution.
To avoid engine damage, the variation speeds of all \( m \) actuators are constrained individually by nonnegative constants \( \Delta_a \) for \( a \in [m] \). Let \( d^q \in S_{f,t} \) and \( d' \) be a data point in either neighboring stack \( S_{f\pm 1,t} \) or \( S_{f,t\pm 1} \). As our drivability constraints we require

\[
s^q + s' \leq 1 \quad \text{if} \quad |[d^q]_a - [d']_a| \geq \Delta_a \quad \text{for any} \quad a \in [m].
\]

(11)

Note that this is a secant constraint, since the data points compared are contained in neighboring stacks. The emission condition (5) describes the upper bound of several emission test cycles, e.g., maximal \( NO_x \) production. Additionally, the current output of a pollutant \( p \in E \) is restricted by the boundaries of the respective interval of noncritical values

\[
J_p = [e_p, \bar{e}_p].
\]

In the following, the weights \( \omega_{ft} \) represent the mean resistance time of the of the rectangles \( OP_{ft} \) in the given test cycle. For simplicity of notation, the mean resistance time on the rectangle corresponding to the stack \( S^q \) containing \( d^q \) is denoted by \( \omega^q \). Then the emission constraint can be formulated as follows:

\[
\sum_{q=1}^{M} \omega^q [d^q]_p s_q + \sum_{(f,t) \in [k] \times [k]} \omega_{ft} \tau_p s_{ft} \leq \bar{e}_p \quad \text{for all} \quad p \in E.
\]

(12)

The second summand of the left hand side of the inequality serves as a penalty term to compensate for stacks that contribute no data point to the solution SOL.

5. AVL Engine Model

For our experiments in Section 6 we replace the test-bench with an engine model of a diesel engine with turbo charger, pilot injection and variable turbine geometry. The latter has been developed in cooperation with AVL GmbH and is based on measurements on a compression ignition/diesel engine. Below we will give a brief overview of the effects of the 8 actuators and 18 measurands that are simulated, thus indicating the scope of the simulation. For a detailed description of the AVL model’s derivation, see [Bur15] and [ea13].

5.1. Actuators of the AVL Engine Model.

Revolution frequency of the crank shaft (RF). The crankshaft converts the reciprocating motion of the cylinders into a rotational motion. In modern four-stroke engines every cylinder fires once for every two revolutions of the crankshaft. The revolution frequency is a controlled actuator, as discussed in Section 3. It stands out among the other actuators since it provides one coordinate axis of the operation field.

Injected fuel quantity (IF). In contrast to a spark-ignited engine, the injected amount of fuel \( Q \) is the most important actuator. More precisely, the injection process is crucial in the application process of diesel engines. The injection process is given by several pre/pilot-injections, a main injection and post-injections. Typically, the engine torque is mainly determined by
Table 2. Actuator-intervals during various calibration runs. LT/HT = low/high torque, FR = free variation of torque; cf. Sections 6.1 and 6.2.

| Act. | Unit       | Basic, LT       | Basic, HT       | Basic, FR       | Full Calibration       |
|------|------------|-----------------|-----------------|-----------------|------------------------|
| RF   | min        | 1000–2600       | 1000–2600       | 1000–2600       | 1000–2600              |
| IF   | mm cycle   | 6–10            | 50–60           | 6–60            | 6–60                   |
| RP   | hPa        | 295677 / 405677 | 295677          | 295677          | 295677–1126537         |
| AF   | mg stroke  | 300             | 300             | 300             | 275–991                |
| TG   | int        | 30              | 30              | 30              | 30–85                  |
| MT   | ° CA       | 0 / 10          | 0               | 0               | 0–10                   |
| PI   | mm³ cycle  | 1               | 1               | 1               | 1                      |
| PT   | µs         | 1540            | 1540            | 1540            | 1540–2565              |

the main-injection $Q$. This actuator defines the total amount of fuel per cycle. In this simple model, the injected fuel volume is divided into one pilot and the main injection.

*Pressure in the common rail system (RP).* In contrast to solenoid-controlled unit injector elements, the pressure is generated by a central fuel and high pressure pump. The fuel injectors are opened and closed by piezo elements. *Air filling (AF).* Similar to a spark ignited engine, an air valve controls the amount of air which contributes to the combustion process. In this model the amount of air is given directly in mass per piston stroke.

*Turbine geometry (TG).* Modern turbochargers do not have a static turbine geometry. Variable-turbine-geometry turbochargers are able to tune the angle of the turbine blades in order to increase the amount of boost. Alternative setups are given by static turbines with waste gates. Waste gates are applied to reduce the amount of exhaust gas that accelerates the turbine, so the amount of boost can be controlled. In our model the geometry is given as a value between 30 and 85.

*Main timing (MT).* The main timing is comparable to the spark timing of Otto-engines. It defines the start timing of chemical reactions in the crankshaft angle of the main injection. Similar to the Otto engine, the pressure rise is delayed by the ignition delay.

*Pilot injection (PI) and pilot timing (PT).* Pilot injection works in tandem with pilot timing to achieve a complete burning of the fuel, which in turn also drastically reduces the emission of $NO_x$ gases. The pilot injection increases the temperature of the combustion chamber, thus when the main injection occurs the fuel is sent into a chamber which already is at a higher temperature than its autoignition point. This especially facilitates the fuel burning at lower speeds.

5.2. Measurands of the AVL Engine Model.

*Torque.* The produced torque of the engine. The revolution frequency and the engine torque define the power level of the ICE.
Fuel mass flow. The amount of fuel which is delivered to the cylinder per hour.

Carbon monoxide, hydrocarbons, nitrogen oxides, soot. The momentary and integral exhaust emission of pollutants, given in parts per million and emission per hour.

Indicated mean pressure. The average pressure over a cycle in the cylinder.

Lambda. Lambda displays the chemical partitioning of the fuel. Spark-ignited engines run on lambda around 1. The combustion limits are 0.6 and 1.6. The lambda value for compression ignition engines is much higher, up to 20.

Manifold pressure. The manifold pressure measures the absolute pressure in front of the intake channel. The pressure depends on the absolute environmental pressure and the boost level of the turbocharging system.

Boost pressure. The boost pressure is created by the turbocharging unit. It depends on the turbine geometry setting and the current combustion behavior.

Maximal cylinder pressure. The maximal point of the cylinder pressure sequence. Every engine has a specified maximal cylinder pressure, in order to avoid damaging of the devices. Typically, the maximal pressure is about 160 bar.

Manifold temperature. The manifold temperature measures the temperature in the intake channel.

Critical temperature. The critical temperature is defined as the temperature of the burn zone when the exhaust valves opens. In case of bad timings, the fuel has not been consumed completely, which results in the release of flames to the exhaust manifold, catalysts and turbocharger. It indicates damages to sensitive parts of the engine setup.

Specific fuel consumption. The current power level of the engine over the current fuel consumption.

6. Optimizing the AVL Engine Model

We subdivided the engine optimization into two phases. During the first phase, which we dub basic calibration, the emission constraint (2) is ignored. The purpose of this omission is to obtain a coarse picture of the engine behavior and derive a preliminary solution that conforms to the drivability constraint (11). In the second phase, which we dub full calibration, a solution conforming to all constraints is obtained by building on and refining the preliminary solution of the first phase.

All measurement times are given as real-world measurement times as opposed to runtimes of the simulation which are highly hardware specific.

6.1. Basic Calibration. During basic calibration, only two actuators are dynamic, IF and RF. The other six actuators are static throughout. The phase is subdivided into three runs of several hours. By a “run” we mean the following: The intervals of the dynamic actuator IF and the values of the static actuators are reset. Then the algorithm steps in Sections 4.2–4.4 are repeated for a prescribed amount of time. The revolution frequency is set to vary over its full range of 1000–2600 revolutions per minute throughout all three runs. For a full account of the applied settings, see Figure 2.
A good foundation for an optimization is a well-defined low and high torque boundary region. In a compression ignition engine the generated torque is roughly proportional to the injected fuel. In the first run we measure the low torque region by limiting the IF-interval to 6–10 mm$^3$/cycle. The static actuators are set to values that support the creation of low torque operation points. The measurement time of this run is 6 hours. The low torque region is measured for a second time span of 6 hours with slightly modified settings, i.e., $MT = 10.0$ and $RP = 405677$ hPa.

Afterwards, the IF-interval is reset to 50–60 mm$^3$/cycle, so that the high torque regions can be measured. The static actuators are set to values that support high torque operation points. Measurement time is again 6 hours.

Finally, the lower and upper bound of IF are removed, in order to get a picture of the remaining region of the solution map. Therefore, the static actuators are set to midrange values. This region is measured for 6 hours, too. After 24 hours, we get a coarse picture of the engine behavior.

Using the data as input for an integer linear program as described in Section 4.6 yields a solution map that still has holes. These are caused by the drivability constraint. These are closed by interpolation/extrapolation as described in Section 4.5. For the comparison of a solution that respects the drivability constraint with one that does not, see Figure 5.

In the current example run it took an additional 45.8 hours to close all measurement holes. The average fuel consumption for 100 kilometers of the so-derived preliminary solution is 4.65 liters. Since this optimization does not take exhaust emissions into account, the cycle integrals of CO, HC and NO$_x$ are rather high: 3.68, 1.72 and 5.26, respectively, per NEDC. The solution map obtained in the first calibration phase serves as the basis of the full calibration.

Figure 4. Operation field saturation after basic calibration, before closing of gaps.
Figure 5. Top: Solution for actuator $AF$ without drivability constraint. Bottom: $AF$ settings of EURO 5 solution respecting the drivability constraint.

6.2. Full Calibration. During the full calibration phase the only static actuator is pilot injection. All others range over their whole respective domains. There is no predefined time limit for the iteration. The algorithm steps in Sections 4.2–4.4 and the solution of the integer linear program in Section 4.6 are repeated until a solution map conforming to the given constraints is derived.

The optimization is performed, first, for the NEDC, then for the RANDOM cycle. An NEDC solution that conforms to the EURO 4 norm is obtained after 68.51 hours, one that conforms to EURO 5 after 88.41 hours. From the data set of the NEDC EURO 5 solution, a EURO 4 solution for the RANDOM cycle can be derived. A EURO 5 solution is obtained after 107.55 hours of measurement. (For the emission constraints corresponding to the different EURO norms, see Table 1.) To lend some context to these numbers,
we performed the same calibrations using uniform grids. To derive a solution of equivalent quality, i.e., a solution conforming to all constraints given by the different EURO norms, the grid approach consistently required 15 to 20 times as many measurements.

Since the AVL model only has a measurand for particle mass, but not for the number of particles, the determination of a EURO 6 solution lies outside of the model’s scope. However, the step from EURO 5 to EURO 6 merely adds an item to the list of emission constraints. This poses no fundamental challenge to our method which is scalable with respect to the number of pollutant limits. (Naturally, adding further constraints will increase the measurement time though.)

**Description of Figures.** Figures 6 and 7 illustrate a typical feature of engine calibrations fitted to specific driving cycles. These essentially subdivide the solution map into two parts. One covered by the cycle where emissions are optimized, and one where they are not. This leads to emission values being 20–100 times higher on the part of the solution map that is not covered by the driving cycle. The NEDC covers only a fraction of the operation field, while the RANDOM cycle covers more than half. This results in lower $\text{NO}_x$ output on the better part of the operation field for the RANDOM calibration; cf Figure 8. Conversely, in Figure 9 one can observe that the calibration solution for the RANDOM cycle displays 10–15% higher values for specific fuel consumption on most points of the operation field than the corresponding operation points of the NEDC.

One can, of course, enforce the selection of data points with lower emission values on the part of the operation field not covered by a driving cycle as well, but at the expense of a higher fuel consumption.

**7. Conclusion**

In this article, we have described a semi-automatic approach to calibrate and optimize internal combustion engines with several actuators and sensors. The side constraints are limits given by safety or technical requirements, bounding the variation speed of actuators and ensuring emission bounds on given driving cycles. Our method automatically performs refinements of measurements, thus focusing the effort of the measurements around regions of nonsmooth behavior of the engine. That is, it automatically identifies neighborhoods of anomalous engine behavior and maps them in appropriate detail. For the so-obtained data an optimal calibration solution is computed.

The output of the algorithm is a solution map which consists of actual measurements and thus reflects the exact behavior of the engine for the given settings, as opposed to indirectly derived actuator setting obtained via modeling or interpolation. This results especially in improved values for pollutant emission and fuel consumption near nonsmooth regions of the admissible domain.

In our experiments, we demonstrated the practicability of the adaptive meshing methodology, showing a significant speed-up in the measurement time in comparison to uniform grids. Moreover, the resulting solutions respect the emission constraints of EURO 4 and 5 norms. We would like to stress that in our method it is easy to take into account further emission
Figure 6. Top: Weighting table of the NEDC. Middle: Resulting NO\textsubscript{x} emissions (in g/h) after calibration for the NEDC. Bottom: NO\textsubscript{x} emissions with logarithmic scale.
Figure 7. Top: Weighting table of the RANDOM cycle. Middle: Resulting NO\textsubscript{x} emissions (in g/h) after calibration for the RANDOM cycle. Bottom: NO\textsubscript{x} emissions on a logarithmic scale.
constraints such as, e.g., the number of emitted particles for the EURO 6 norms.

The next interesting step will be to test the method on an actual combustion engine. It is our expectation that the experimental findings of this work will transfer well to the real-life setting which is the engine test bench. That is, we expect the concept of adaptive grid refinement to considerably improve on uniform grids in terms of the number of measurement required to obtain a sufficiently meaningful image of the engine behavior to derive calibration settings respecting a given set of constraints from the obtained data.

**References**

[AI18] Alexandr Andoni and Piotr Indyk. Nearest neighbors in high-dimensional spaces. In Csaba D. Tóth, Jacob E. Goodman, and Joseph O’Rourke, editors, Handbook of Discrete and Computational Geometry, chapter 43. CRC Press, 2018. 3rd edition.

[Bur15] Timo Burggraf. Development of an automatic, multidimensional, multicriterial optimization algorithm for the calibration of internal combustion engines. PhD thesis, TU Darmstadt, 2015.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Third Edition. The MIT Press, 3rd edition, 2009.

[Coo98] William Cook. Combinatorial optimization. Wiley, New York, 1998.

[ea13] M. Vogels et al. DoE Model “Compression ignition engine”. AVL LIST GMBH, Graz, 2013.

[FEZ04] FEZ. Topexpert, 2004. http://www.fev.com/en/what-we-do/software-and-testing-solutions/products/testing/calibration/topexpert-suite.

[GPFL01] Kurt Gschweitl, Horst Pfluegl, Tiziana Fortuna, and Rainer Leithgoeb. Steigerung der Effizienz in der modellbasierten Motoren-Applikation durch die neue CAMEO Online DoE-Toolbox. ATZ – Automobiltechnische Zeitschrift, 103(7):636–643, Jul 2001.
Figure 9. Actuator IF after calibration for the NEDC (top) and RANDOM cycle (middle). Bottom: Operation points with higher values in calibration for RANDOM cycle than for NEDC. Colors represent ratio of consumption values.
[Ise10] Rolf Isermann. Elektronisches Management motorischer Fahrzeugantriebe. ATZ/MTZ-Fachbuch. Vieweg+Teubner, Wiesbaden, 2010.

[KKL10] T. Kruse, S. Kurz, and T. Lang. Modern statistical modeling and evolutionary optimization methods for the broad use in ECU calibration. IFAC Proceedings Volumes, 43(7):739–743, 2010. 6th IFAC Symposium on Advances in Automotive Control.

[Kos03] Kosmas Knoedler et. al. Modellbasierte Online-Optimierung moderner Verbrennungsmotoren. MTZ – Motortechnische Zeitschrift, 64(6):520–526, Jun 2003.

[M+16] Inge Mayeres et al. Transitions towards a more sustainable mobility system. Technical Report EEA Report No. 34/2016, European Environment Agency, 2016. https://www.eea.europa.eu/publications/term-report-2016.

[Mit00] Alexander Mitterer. Optimierung vielparametriger Systeme in der KFZ-Antriebsentwicklung. PhD thesis, TU München, Düsseldorf, 2000.

[RKNS07] Karsten Roepke, Mirko Knaak, Adrian Nekler, and Steffen Schaum. Rapid measurement. MTZ – Motortechnische Zeitschrift, 68(4):276–282, Apr 2007.

[Sch98] A. Schrijver. Theory of linear and integer programming. John Wiley & Sons, 1998.

[SHI00] Matthias Schueler, Michael Hafner, and Rolf Isermann. Einsatz schneller neuronaler netze zur modellbasierten optimierung von verbrennungsmotoren. MTZ – Motortechnische Zeitschrift, 61(10):704–711, Oct 2000.

[Sie01] Gerard Sierksma. Linear and Integer Programming: Theory and Practice. CRC Press, 2nd edition, 2001.

Bertrandt Technikum GmbH, 71139 Ehningen, Germany
E-mail address: timoburggraf@gmx.de

Workgroup Discrete Mathematics/Geometry, Technische Universität Berlin, 10623 Berlin, Germany
E-mail address: joswig@math.tu-berlin.de

Research Group Optimization, Technische Universität Darmstadt, 64293 Darmstadt, Germany
E-mail address: pfetsch@mathematik.tu-darmstadt.de

Workgroup Discrete Mathematics/Geometry, Technische Universität Berlin, 10623 Berlin, Germany
E-mail address: radons@math.tu-berlin.de

Research Group Optimization, Technische Universität Darmstadt, 64293 Darmstadt, Germany
E-mail address: ulbrich@mathematik.tu-darmstadt.de