

Semi-automatically optimized calibration of internal combustion engines

Timo Burggraf · Michael Joswig · Marc E. Pfetsch · Manuel Radons · Stefan Ulbrich

Received: date / Accepted: date

Abstract Modern combustion engines incorporate a number of actuators and sensors that can be used to control and optimize the performance and emissions. We describe a semi-automatic method to simultaneously 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 incorporates 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.

Keywords Internal Combustion Engines · Calibration · Mixed Integer Nonlinear Optimization

Research by M. Joswig is partially supported by Einstein Stiftung Berlin and Deutsche Forschungsgemeinschaft (EXC 2046: “MATH+”, SFB-TRR 109: “Discretization in Geometry and Dynamics”, SFB-TRR 195: “Symbolic Tools in Mathematics and their Application”, and GRK 2434: “Facets of Complexity”). 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”.

Timo Burggraf
Bertrand Technikum GmbH, 71139 Ehningen, Germany
E-mail: timoburggraf@gmx.de

Michael Joswig
Institut für Mathematik, MA 6-2, Technische Universität Berlin, 10623 Berlin, Germany
E-mail: joswig@math.tu-berlin.de

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

Manuel Radons
Institut für Mathematik, MA 6-2, Technische Universität Berlin, 10623 Berlin, Germany
E-mail: radons@math.tu-berlin.de

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

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. [14].

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 frequency, 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 involved and not known exactly a priori, i.e., it has to be measured and approximated. 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.

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 (ECU). The settings on the engine manifold have to be chosen in such a way that they are consistent across various 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 are supposed to 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 in relevant 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, based on the so-obtained information about the engine behavior, one should optimally choose the resulting engine manifold and solution map. More precisely, one should select actuator settings for the final solution map that obey the above-mentioned restrictions and yield an approximation to an optimal solution map which is sufficiently close in the sense that all prescribed emission targets are met.

In this article we propose a new 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 is not. This leads to a more accurate representation of the engine manifold than with a uniform grid approach with a fraction of the measurements. This first part is based on the well-established LOLIMOT (local linear model tree) [19]

partitioning scheme of the space of actuator settings, which we extend by more involved measurement-routing and grid-refinement schemes.

- *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 briefly review 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 the AVL Engine Model, cf. see [2, p. 69ff] and [21], as a practical case-study, and Section 6 contains the experimental results. We close with a few 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 measurement and calibration. In the following we give a brief overview of the main approaches. Any calibration process comprises the following two major components:

- (A) Obtain a good approximation of the engine behavior, mathematically described in terms of some function.
- (B) Use whatever (approximate) knowledge of that function to produce a set of lookup tables for the ECU.

The lookup tables may be optimized for various goals, e.g., dynamic driving behavior or maximal performance. In the present work we are set on optimizing the fuel consumption while conforming to a set of emission constraints.

2.1 A Naïve Measurement-Based Approach

A basic idea is to measure all actuator settings on a sufficiently fine uniform grid. For modern engines this approach is infeasible for two reasons. First, the size of the grid increases exponentially with the number of actuators. This would increase the number of measurements—and thus also the measurement time—beyond any feasible bounds. Second, even if such a comprehensive measurement were possible, it is computationally infeasible to optimize over such large input. So neither Step (A) nor Step (B) can be realized in this way.

2.2 The Model-Based Approach

For Step (A) in a model-based calibration the measurements are used to fit a given model to the engine behavior. Once this fitting process is completed, in Step (B) lookup tables for the ECU which are optimal with respect to the so-obtained model and a given set of objectives are computed via standard techniques from nonlinear optimization such as steepest descent methods, cf. [9, p. 542]. A software package which does so in an automated fashion is the *Model-Based Calibration Toolbox* for MATLAB [13]. Leading model-based approaches employ physical models, the training of neural networks via the measurements, and statistical machine learning. An implementation that combines the first two kinds is, e.g., the software package `mbminimize` [10]. Statistical machine learning is used, e.g., in ASCMO [11].

Physical models are usually described in terms of smooth functions. Similarly, most neural networks also model smooth functions. For instance, perceptrons are commonly composed of sigmoid functions and thus smooth, cf. [7, pp. 103–111]. This restriction to mathematical modeling via smooth functions inherently imposes severe limitations to any model-based calibration of ICEs. The reason is that the functions necessary to describe an ICE well enough exhibit strong nonlinearities and may even be nondifferentiable. In the former case, the approximation error can be bounded globally, but locally the approximating functions are likely to exhibit strong oscillations that do not describe the underlying engine behavior very well; cf. [7, p. 105]. At a nondifferentiability of a continuous function the role of the local linear approximation, which is the derivative, is taken over by local piecewise linear approximations [17, p. 67ff]. In this case any sufficiently good approximation usually requires an exponential number of local models, one for each linear piece. As a consequence, it is common that either Step (A) does not describe the ICE well enough or both Step (A) and Step (B) are infeasible due to a combinatorial explosion, just like in the naïve approach.

2.3 LOLIMOT

Local linear model trees (LOLIMOT) are arguably the most performant model-based calibration method to date; cf. [9, p. 93], as well as [12]. LOLIMOT partitions the space of actuator settings as a dissection into cubical cells. In each cubical cell, the engine behavior is modeled by a Gaussian function on the basis of a “central composite measurement point pattern”; cf. [9, Fig. 3.4.12]. These local models are then stitched together into a global one. If a local model fails to produce a sufficiently close approximation, e.g., due to a strongly nonlinear behavior of the engine, the corresponding cubical cell is split along one of its axes and the engine behavior is modeled via a central composite design on each sub-cell. This process is iterated until a sufficient model quality is reached globally.

The key contribution of LOLIMOT is to introduce adaptive meshing to the engine calibration process. One drawback of the method is that only one output value is modeled at a time. The engine model is then fused together from the component models for the individual measurands; cf. [9, p. 93]. This approach may cause both holes in the modeled behavior as well as local redundancies of data. Further, the model produced by LOLIMOT is a special type of radial basis function network; cf. [8, p. 143]. These types of neural nets require a locally homogeneous covering of the actuator space by measurements; cf. [7, p. 111]. This then necessitates the aforementioned fixed measurement patterns within each cell. But any such fixed pattern is again, like in the previous approaches, subject to a combinatorial explosion in high dimensions. The adaptive meshing with fixed local pattern is better than the uniform grid used in the naïve approach, but only by a constant factor. This is the second drawback.

Our new method for engine calibration refines LOLIMOT by simultaneously considering all measurands. This employs a more involved grid refinement scheme and randomized measurement routing; cf. Section 4.2. In this way we can overcome conceptual limitations of LOLIMOT.

2.4 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 destruction, e.g., for cylinder pressures or critical device temperatures. Others are induced, e.g., by emission and noise regulations. In practice they are obtained automatically via established software packages such as Cameo [6], TopExpert [5], or the MATLAB toolbox LOLIMOT (local linear model tree) [19]. Throughout we will assume that all such boundary informations are already given.

3 Mathematical Problem Description

The calibration of an internal combustion engine is the procedure to derive an engine manifold, which is optimal with respect to some predefined objective. 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). The calibration problem posed informally in [15, p. 250] asks for an engine manifold that minimizes fuel consumption, while conforming to a number of emission constraints. In this section we will present two mathematical formalizations of the latter, one idealized continuous version and its discretization whose output fits the format of the lookup tables for the ECU.

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.

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 $U_{\text{ad}} \subset \mathbb{R}^m$, which we call the *admissible domain*. It was already noted in Section 2.4 that throughout this work we assume U_{ad} to be already given. The noncritical sensor values define another box $Y_{\text{ad}} \subset \mathbb{R}^n$, the *admissible range*; cf. Section 2.4. 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

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

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

$$\text{OP} := \{(u_{\text{freq}}, y_{\text{torq}}) \mid (u, y) \in F_{\text{fsb}}\}$$

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

It is important to understand that the actual optimization problem is inherently discrete, since the desired output is a solution map for the ECU. These lookup tables of actuator values for given frequency and torque combinations have a given finite length. For the sake of a concise exposition, however, we now describe an idealized continuous optimization problem which has the actual optimization problem that we want to solve as a natural discretization. While similar optimization problems must be behind all known approaches to ICE calibration, we are not aware of any complete description in the available literature. Our model below is intended to fill this gap.

Optimization Space and Drivability

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, i.e.,

$$[F(M(u_{\text{freq}}, y_{\text{torq}}))]_{\text{torq}} = y_{\text{torq}} \quad \text{for all } (u_{\text{freq}}, y_{\text{torq}}) \in \text{OP} . \quad (1)$$

A basic requirement is that these maps are continuous.

Moreover, there are vital additional conditions to consider. Any solution to the engine calibration problem must be drivable in the sense of [15, p. 258]: 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 Δ_a for $a \in \{1, \dots, m\}$. For a given map M and actuator a , the corresponding *drivability constraint* is that for all $(u_{\text{freq}}, y_{\text{torq}}), (u'_{\text{freq}}, y'_{\text{torq}}) \in \text{OP}$ with $u := M(u_{\text{freq}}, y_{\text{torq}})$ and $u' := M(u'_{\text{freq}}, y'_{\text{torq}})$ the following has to hold

$$|u_a - u'_a| = |[M(u_{\text{freq}}, y_{\text{torq}})]_a - [M(u'_{\text{freq}}, y'_{\text{torq}})]_a| \leq \Delta_a \cdot \|(u_{\text{freq}}, y_{\text{torq}}) - (u'_{\text{freq}}, y'_{\text{torq}})\|, \quad (2)$$

where $\|\cdot\|$ is some norm. As a consequence, the map $M_a: \text{OP} \rightarrow \mathbb{R}$ is Lipschitz continuous with constant Δ_a . We define Ω as the set of all maps M which are feasible in the sense that they obey the drivability constraint with respect to each actuator, more precisely,

$$\Omega := \{M: \text{OP} \rightarrow \mathbb{R}^m: M \text{ satisfies (1) and moreover (2) for all } a \in \{1, \dots, m\}\}.$$

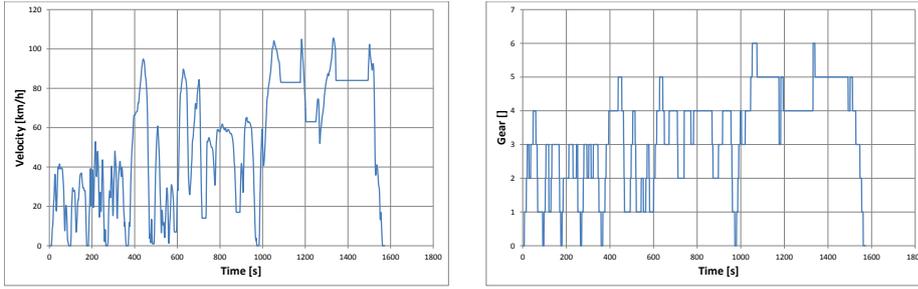


Figure 1 Operational profile of RANDOM driving cycle.

Driving Cycle and Emission Constraints

In accordance with current government regulations and common test cycles, the engine behavior is optimized with respect to pre-defined scenarios, known as driving cycles. In our continuous model a *driving cycle* is a time-parametrized curve

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

whose purpose is to simulate the phases of acceleration and constant speed of real-world driving patterns. Driving cycles are given indirectly in the form of operational profiles which map a time to a gear/velocity combination; cf. Figure 1. Taking into account the weight of the car, its drag coefficient, specific tire friction, etc., every such gear/velocity combination can be mapped to a revolution frequency/torque combination in the operation field.

For the sake of the simplicity of exposition, here we will assume that the calibration is performed with respect to a fixed driving cycle. 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 (NO_x) as well as particulate matter (PM) and number (PN). We denote by E the index set that corresponds to pollutant emissions and by ϵ_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 γ_{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 $p \in E$:

$$\int_0^1 [F(M(\gamma_{dc}(t)))]_p \cdot \|\dot{\gamma}_{dc}(t)\| dt \leq \epsilon_p. \quad (3)$$

The factor $\|\dot{\gamma}_{dc}(t)\|$ accounts for the fact that γ_{dc} is not necessarily normalized, i.e., the speed of acceleration, deceleration, etc., varies. Table 1 shows the diesel engine emission constraints for EURO norms 3-6c (E3-E6c).

The optimization formulation

Let “fuel” be the index of the actuator for the injected fuel quantity. It is possible that one wants to optimize fuel consumption over a different curve than the driving cycle γ_{dc} . We

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
NO _x	mg/km	150	80	60	60	60
PM	mg/km	–	–	5	4.5	4.5
PN	1/km	–	–	–	6 · 10 ¹²	6 · 10 ¹¹

thus define a second curve in the operation field

$$\varphi: [0, 1] \rightarrow \text{OP}.$$

Using the terminology developed so far, the continuous optimization problem is

$$\begin{aligned} & \text{minimize}_{M \in \Omega} \int_0^1 [M(\varphi(t))]_{\text{fuel}} \cdot \|\dot{\varphi}(t)\| dt & (4) \\ & \text{subject to} \int_0^1 [F(M(\gamma_{\text{dc}}(t)))]_p \cdot \|\dot{\gamma}_{\text{dc}}(t)\| dt \leq \epsilon_p \quad \text{for all } p \in E. \end{aligned}$$

That is, the optimization goal is to minimize the total consumed fuel with respect to the time parametrized curve φ in OP, while for all pollutants p the integral emission along γ_{dc} is bounded by the prescribed constant ϵ_p . Implicitly, problem (4) is subject to the drivability constraint, due to the fact that all manifold maps $M \in \Omega$ must conform to inequality (2). In the discrete formulation of the engine calibration problem this dependence will be made explicit. We further remark that the curves φ and γ_{dc} may coincide, but they do not have to. More elaborate curves φ may be useful, e.g., for controlling the fuel consumption also outside the driving cycle, while ignoring the fringes of the operation field.

3.2 Discrete Optimization Problem

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 and $t \in [k] := \{1, \dots, k\}$ denote the rectangle's frequency and torque coordinate, respectively. This corresponds to the technical requirement that 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 M 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 $u \in U_{\text{ad}}$ with $(u, F(u)) \in F_{\text{fsb}}$ such that the pair $(u_{\text{freq}}, y_{\text{torq}})$ 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 non-linear 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.

Splitting the engine calibration into Steps (A) (data acquisition) and (B) (computation of a solution map), cf. Section 2, our goal can be formulated as follows. In Step (A) obtain, via actual measurements of the engine, a finite set

$$\widehat{F} := \{(u^1, y^1), (u^2, y^2), \dots, (u^\delta, y^\delta)\}$$

of δ points in F_{fsb} , i.e., a set of actuator settings $u^q \in U_{\text{ad}}$ such that $y^q = F(u^q) \in Y_{\text{ad}}$, which is a sufficiently good representation of F . Here “sufficiently good” means that we can, in Step (B), extract from \widehat{F} a solution map that conforms to discretized versions of the drivability and emission constraints presented in Section 3.1. In this sense, a solution map is a sufficiently good discrete approximation to an optimal engine map M of the continuous optimization problem (4).

Since \widehat{F} is obtained via actual measurements, it is called the *data set*. The elements

$$d^q := (u^q, y^q)$$

of \widehat{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 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 \widehat{F} into sets

$$S_{ft} := \{(u, y) \in \widehat{F} \mid (u_{\text{freq}}, y_{\text{torq}}) \in \text{OP}_{ft}\}.$$

We call each set S_{ft} a *stack*, and the entire partition

$$k\text{OP} := \{S_{ft} \mid f, t \in [k]\}$$

is the *k-operation field* of F with respect to \widehat{F} . In the solution map each entire rectangle OP_{ft} will be represented by a single measurement $d_{ft} \in S_{ft}$. Hence, the discrete analogue to Ω , the set of all manifold maps, is the set Ω_k of all maps $M_k: [k] \times [k] \rightarrow \mathbb{R}^m \times \mathbb{R}^n$, such that $M_k(f, t) \in S_{ft}$, i.e., select exactly one data point from each stack, and satisfy a discrete analogue of the drivability constraint (2), see (5) below. Thus,

$$\Omega_k := \{M_k: [k] \times [k] \rightarrow \mathbb{R}^m \times \mathbb{R}^n: M_k(f, t) \in S_{ft} \text{ for all } f, t \in [k] \text{ and } M_k \text{ satisfies (5)}\}.$$

The elements of Ω_k are called solution maps. Each solution map $M_k \in \Omega_k$ has precisely k^2 values $M_k(f, t) \in S_{ft}$, one per stack. A solution map is uniquely determined by the data points $d_{ft} = M_k(f, t)$, $f, t \in [k]$, and we call d_{ft} the *representatives* of the k^2 stacks S_{ft} corresponding to M_k . The final calibration solution is a solution map $\text{SOL} \in \Omega_k$ which is optimal with respect to the given objectives, i.e., minimization of fuel consumption, subject to emission regulations and drivability. The final solution will be picked by solving the optimization problem (8) below.

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, with $d_{ft} = M_k(f, t)$, the discrete analogue of (2) is

$$|[d_{ft}]_a - [d_{gs}]_a| = |[M_k(f, t)]_a - [M_k(g, s)]_a| \leq \Delta_a \quad \text{for all } a \in [m], \quad (5)$$

and all tuples $(f, t), (g, s) \in [k] \times [k]$, where either $g = f \pm 1$ or $s = t \pm 1$.

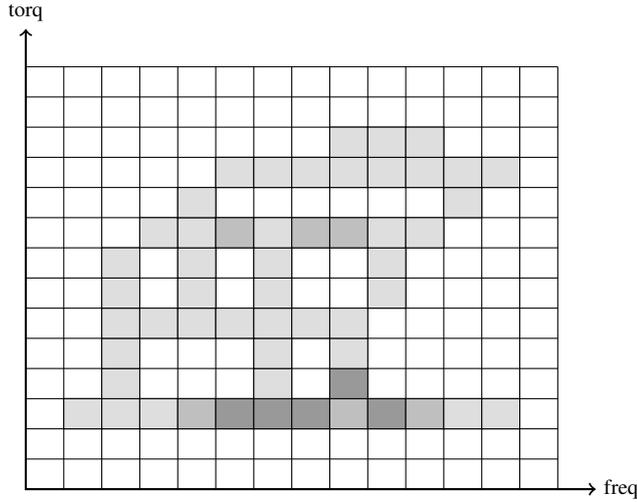


Figure 2 Schematic of a discretized operation field and driving cycle. Resistance times are represented by gray shades. (See also Figures 7 and 8.)

Discrete emission constraint

To discretize the emission constraint (3), we replace an engine map $M \in \Omega$ by its discrete counterpart $M_k \in \Omega_k$ and obtain instead of the curve integral along the curve γ_{dc} a weighed sum. In fact, to each rectangle OP_{ft} we associate a weight ω_{ft} . This weight is set to zero if the intersection of OP_{ft} with the image of the curve γ_{dc} is empty. Otherwise ω_{ft} is a positive value that reflects the resistance time, i.e., the duration of the curve γ_{dc} staying in the rectangle OP_{ft} . The function

$$\text{DC}: [k] \times [k] \rightarrow \mathbb{R}_{\geq 0}, (f, t) \mapsto \omega_{ft} \quad (6)$$

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

The *discrete emission constraint* is now given as

$$\sum_{(f,t) \in [k] \times [k]} \omega_{ft} \cdot [M_k(f,t)]_p \leq \epsilon_p \quad \text{for all } p \in E. \quad (7)$$

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 1 and Section 3.1. The weights ω_{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. For a schematic of a discretized driving cycle, cf. Figure 2.

Discrete optimization formulation

In the discrete version of problem (4), we approximate the objective function in a similar fashion by replacing again the integral along the curve φ by a weighted sum. To this end, let as above the weight function

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

be the discrete analogue of the curve φ describing how long the curve φ stays in OP_{ft} . Again, Φ may coincide with DC from (6) or be chosen to optimize fuel consumption on a larger part of the operation field. The optimization objective is now to find a solution map $\text{SOL} \in \Omega_k$ by picking for each stack S_{ft} a single representative $d_{ft} = \text{SOL}(f, t)$, so that $\text{SOL} \in \Omega_k$ solves

$$\text{minimize}_{M_k \in \Omega_k} \sum_{(f,t) \in [k] \times [k]} \Phi_{ft} \cdot [M_k(f, t)]_{\text{fuel}} \text{ subject to the emission constraint (7)}. \quad (8)$$

Note that $M_k \in \Omega_k$ includes the discrete drivability constraint (5). In Section 4.5 we will demonstrate how to formulate this problem as an integer linear program (ILP).

4 Semi-automatic Calibration

Our method consists of two parts, one semi-automatic, and one automatic. Algorithm 1 is a first rough sketch of the automatic part, which will be detailed in the following. 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 solution map $\text{SOL}: [k] \times [k] \rightarrow \mathbb{R}^m \times \mathbb{R}^n$, $\text{SOL} \in \Omega_k$, of representatives of the k -operation field. In practice, the evaluation oracle for F is given by an engine mounted on a test-bench.

The while loop of Algorithm 1, whose description makes up the bulk of this section, terminates if a preliminary solution map $\widetilde{\text{SOL}}: [k] \times [k] \rightarrow \mathbb{R}^m \times \mathbb{R}^n$ could be extracted from the measured data. This preliminary solution map $\widetilde{\text{SOL}}$ conforms to the emission and drivability constraints. However, for each stack S_{ft} in k OP, it contains either a data point d_{ft} , which will then represent S_{ft} , or a placeholder that adds penalties to the total emission; cf. Section 4.5. Note that the constraint $M_k \in \Omega_k$ in (8) requires, that there are no empty stacks, but our integer linear program (ILP) formulation of (8) in 4.5, which is used by Algorithm 1, is extended such that it can handle empty stacks by using a placeholder instead. In the second part, the gaps in $\widetilde{\text{SOL}}$ (marked with placeholders) are closed via interpolation-guided measurements and a complete solution map $\text{SOL} \in \Omega_k$ is returned; cf. Section 4.6. The penalty values ensure that replacing a placeholder by a measured value can only improve the solution. As the preliminary solution map $\widetilde{\text{SOL}}$ conforms to the given emission constraints, so must the complete solution map SOL , which is thus a solution to problem (8).

In the semi-automatic part of our method, engineering knowledge is used to provide a meaningful base set of measurements for Algorithm 1 and thus guide the calibration process. During this so-called basic calibration, the first two steps in Algorithm 1 are repeated for a preset amount of time with several actuators fixed to values that enforce measurements in regions which are known to be critical to any calibration effort, regardless of the specific engine. In particular, low torque regions, which have a significant influence on the overall pollutant emission, are focused on hereby. As this semi-automatic part of our method consists of a subset of the steps in Algorithm 1, we did not dedicate a separate section to its description. Instead, we provide a protocol of the actuator settings and their respective purposes during basic calibration in Section 6.1.

Algorithm 1: Engine calibration procedure

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

Output: solution map SOL, updated data set \widehat{F}

initialization

while no ILP-solution $\widetilde{\text{SOL}}$ found **do**

iteration step: adds to \widehat{F}

data cleaning: reduces \widehat{F} to \widehat{F}_{red}

grid refinement

check integer linear program from Section 4.5 for feasibility with \widehat{F}_{red} as input and extract solution $\widetilde{\text{SOL}}$ if it exists

end

close the gaps in $\widetilde{\text{SOL}}$

return (SOL, \widehat{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 \widehat{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 \widehat{F} reflect true values of F . The domain

$$U_{\text{ad}} = \prod_{i=1}^m I_i \quad (9)$$

is a product of intervals I_1, I_2, \dots, I_m , where I_i defines the range of variation of actuator i ; cf. Section 2.4. 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 \widehat{F} of data points is empty. Then, for each dynamic actuator i , the corresponding interval I_i is subdivided into parts of equal length. Otherwise, if \widehat{F} is not empty, then we assume that each interval I_i is equipped with its own subdivision. In either case, the product form (9) induces a grid structure G on the domain. If \widehat{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 being damaged. Then

$$Y_{\text{ad}} = \prod_{j=1}^n J_j.$$

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 \widehat{F} . Thus, it may happen that \widehat{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 \widehat{F} , we construct an abstract graph $\mathcal{G} = \mathcal{G}(G, E)$ as follows. The nodes of \mathcal{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 \widehat{F} is empty, then the grid G is uniform, and the graph \mathcal{G} is the dual graph of a cubical cell complex. Due to non-uniform refinement, the structure of \mathcal{G} will become more complicated.

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

$$w_b(B) = \frac{\text{vol}(B)}{\#B + 1}, \quad (10)$$

which we call the *reciprocal data density* of B . Adding 1 in the denominator prevents division by zero. Further, we define the weight w_e of an edge between two adjacent grid boxes B and B' as the *data density* of $B \cup B'$. That is,

$$w_e(B, B') = \frac{\#(B \cup B')}{\text{vol}(B) + \text{vol}(B')}. \quad (11)$$

There is some room for adjusting these weights; the general idea is that the weights should reflect the data densities.

The classical way to measure the engine behavior is to take a measurement only once a steady-state is reached after an adjustment of the actuators. Accordingly, this technique is called steady-state, or stationary measurement. A more recent approach is the quasi-stationary measurement, also called sweep-mapping. Here, the actuators are varied slowly and continuously according to a ramp function in order to save measurement time. The output then follows with a little delay, while measurements are taken at a regular frequency; cf. [9, p. 93]. There are several techniques to bound and even compensate the contouring error of quasi-stationary measurements; cf. [9, p. 94ff] and Remark 1. Throughout the while loop of Algorithm 1 quasi-stationary measurements will be performed exclusively. For these the following concept is crucial.

Definition 1 (Measurement Ramp) For two points $u^q, u^r \in U_{\text{ad}}$, we call the set

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

the *measurement ramp* from u^q to u^r with ℓ measurements.

Zero-entries of $u^r - u^q$ correspond to actuators which are locally static. Let $\delta := |\widehat{F}|$ and suppose that there is an admissible point $u = u^\delta \in U_{\text{ad}}$ 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:

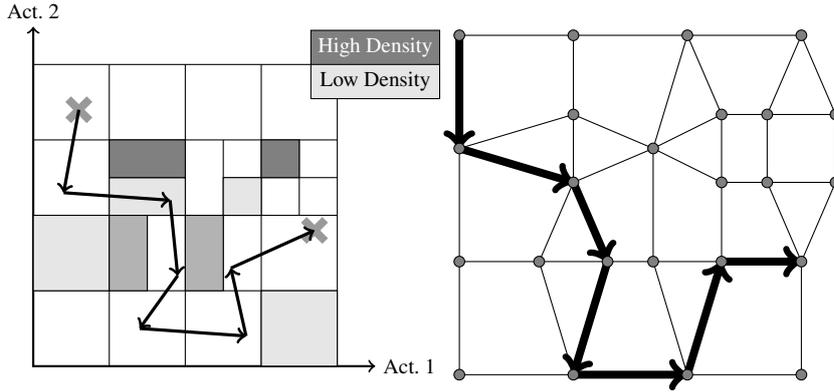


Figure 3 The routing during measurement planning prefers regions of low data density. On the right: Corresponding path in the induced abstract graph.

- I. *Random grid box*: Pick a grid box B' at random with probability

$$\frac{w_b(B')}{W},$$

where $W = \sum_{B \in G} w_b(B)$ is the sum of the reciprocal data densities of all boxes.

- II. *Measurement path*: Determine a shortest path $B = B_0, B_1, \dots, B_s = B'$ in \mathcal{G} from B to B' using Dijkstra's algorithm with respect to the weights w_e in (11); cf. Figure 3 and [3, §2.2]. In each box B_q for $q \in [s]$, pick a point $u^{\delta+q}$ uniformly at random. For $1 \leq j \leq s$, connect the points $u^\delta, u^{\delta+1}, \dots, u^{\delta+s}$ by s measurement ramps with ℓ_j measurements each.

This will result in a total of up to $\sum_{j=1}^s \ell_j$ new measurements to be added to the set \hat{F} . In our setting, the actuators are varied at a constant speed on each measurement ramp. This speed is adjusted so that at least one actuator is varied at its maximal variation speed; the values for the maximal actuator variation speeds are listed in Table 2. The measurement frequency throughout the whole calibration process is set to one measurement per second. As a consequence, both the length and the orientation of the measurement ramps determine the numbers ℓ_j .

Measurement

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 ℓ_j 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 $\sum_{j=1}^s \ell_j$ 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 admissible 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

$$\widehat{F} = \{(u^1, y^1), (u^2, y^2), \dots, (u^\delta, y^\delta)\}$$

is available. In Section 4.4, we will compute local polynomial fits to the function F using data points in \widehat{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 \widehat{F} , since all points lie on a piecewise linear path given by the sequence of measurement ramps. We thus need to extract a subset \widehat{F}_{red} from \widehat{F} that is sufficiently generic.

Furthermore, we want to translate the discrete version (8) 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 4.5 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 $\widehat{U} = \{u^1, u^2, \dots, u^\delta\}$, which is the projection of the set \widehat{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 \widehat{U}_{red} , which yields

$$\widehat{F}_{\text{red}} = \{(u, y) \in \widehat{F} \mid u \in \widehat{U}_{\text{red}}\}.$$

The set \widehat{F}_{red} has significantly fewer data points compared to \widehat{F} . As the threshold for the connection of u^q and u^r by an edge depends on the size of the grid boxes containing them, the distribution of the remaining data points reflects the structure of the grid G . Due to the criteria for grid refinement, which will be introduced in (V) and (VI) below, the structure of the grid G reflects the behavior of F . In particular, this 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.

Note that this reduction does not affect the set \widehat{F} , but we rather explicitly keep \widehat{F}_{red} as a subset. In this way, all our measurements are taken into account for computing the weights according to (10) and (11) in (I) in the next round of measurements.

Remark 1 Since the points u^i are picked in a randomized fashion, the piecewise differentiable map F is differentiable at u^i almost surely. Thus, we may assume that every u^i lies at the center of an open ball on which F is smooth. The grid structure is refined and thus the measurement density increased in areas which are not sufficiently smooth; cf. Section 4.4. Consequently, in an individual grid box one can, with high probability, cluster the measured points according to smooth patches of F . As a result, for such clusters the local time constants are identifiable and the techniques for reversing the contouring error by dynamic correction in [9, p. 95f] can be applied successfully, leading to stored measured values that closely approximate the steady state values of the engine.

4.4 Grid Refinement

The purpose of refining the grid G is to accumulate sufficiently many “meaningful” data points in the set \widehat{F}_{red} , such that the k^2 representatives for the solution map can be extracted while observing the 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_{ad} such that the data point (u^q, y^q) adds significant information to our data set \widehat{F}_{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 $\{u^{q1}, u^{q2}, \dots, u^{qN}\} \subseteq \widehat{U}$. 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., [19]. Our data cleaning method in Section 4.3 ensures poisedness (with high probability).

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 intention is to compare the measured value $y^q \in Y_{\text{ad}}$ at a point $u^q \in U_{\text{ad}}$, where $(u^q, y^q) \in \widehat{F}_{\text{red}}$, with a polynomial interpolation of F in u^q using elements of \widehat{F}_{red} as interpolation points. The error of such an approximation is minimized if we use the N closest neighbors $\{u^{q1}, u^{q2}, \dots, u^{qN}\}$ of u^q in U_{ad} as interpolation points, such that

$$\{(u^{q1}, y^{q1}), (u^{q2}, y^{q2}), \dots, (u^{qN}, y^{qN})\} \subseteq \widehat{F}_{\text{red}}.$$

Concerning the implementation of the polynomial interpolation: As can be seen in the experimental section below, we will usually have less than 100,000 points in \widehat{F}_{red} . This makes the following brute-force approach feasible. Let $\delta_{\text{red}} := |\widehat{F}_{\text{red}}|$ and assume for simplicity that $q = 1$. Now calculate the squared Euclidean distances of $u^2, u^3, \dots, u^{\delta_{\text{red}}}$ to u^1 .

This has a cost of roughly $3 \cdot m \cdot \delta_{\text{red}}$ elementary arithmetic operations. Store these values in the first row of a $(2 \times [\delta_{\text{red}} - 1])$ -array and the indices of the corresponding points u' in the second row ($\approx 2 \cdot \delta_{\text{red}}$ 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. [4, 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 δ_{red} points in \widehat{F}_{red} , we arrive at an approximate cost of $(3 \cdot m + 4) \cdot \delta_{\text{red}}^2 \in \mathcal{O}(m \cdot \delta_{\text{red}}^2)$ elementary operations. In our setting, for up to 100,000 points and 8 dynamic actuators, this results in a total of about 280 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 [1]. 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 (8) of the constrained optimization problem (4), 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 solution map $\text{SOL} \in \Omega_k$ represented by k^2 data points $d_{ft} = \text{SOL}(f, t)$, where $f, t \in [k]$, which cover the k -operation field, while satisfying the emission constraints (7). In practice, it is a major challenge to find sufficiently many points which satisfy the conditions (7) imposed by emission control. This leads us to a second type of grid refinement.

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

$$[F(u^\delta)]_p < \min_{(u^q, y^q) \in \mathcal{S}_{ft}} ([F(u^q)]_p),$$

i.e., if the emission measurement $[F(u^\delta)]_p$ is lower than any other value on the corresponding stack $\mathcal{S}_{ft} \subset \widehat{F}$, 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^δ 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 \widehat{F}_{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^δ, y^δ) with better emission values than those currently stored in \widehat{F} .

4.5 An Integer Linear Program

As advertised in Section 3.2, we will now detail how to formulate the discrete optimization problem (8) as an integer linear program (ILP) in order to derive a preliminary solution map $\widetilde{\text{SOL}}: [k] \times [k] \rightarrow \mathbb{R}^m \times \mathbb{R}^n$ from the reduced set of measurements \widehat{F}_{red} that satisfies the objectives listed in Section 3.2. This ILP-solution $\widetilde{\text{SOL}}$ is not necessarily a complete solution map SOL, as it may contain some placeholders, which must be replaced in a subsequent interpolation step; cf. Section 4.6. For an introduction to the solution of ILPs, see [3], [18], and [20].

For a data point d^q , 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 we call the weight

$$\Phi^q := - \frac{\min\{[d^r]_{\text{fuel}} \mid d^r \in S^q\}}{[d^q]_{\text{fuel}}}$$

the *prey value* of d^q . The prey value of d^q is the negative of the quotient of the minimal fuel consumption among all data points in S^q , the stack containing d^q , over the fuel consumption at d^q . As such, the absolute value of any prey value is smaller than or equal to 1. Ideally, a prey value equals -1 .

Further, for each index $q \in [\delta_{\text{red}}]$, where $\delta_{\text{red}} := |\widehat{F}_{\text{red}}|$, we introduce a binary decision variable $s^q \in \{0, 1\}$, which indicates whether the data point d^q is part of the ILP-solution $\widetilde{\text{SOL}}$. The objective function of our integer linear program can now be written as

$$\text{minimize } \sum_{q=1}^{\delta_{\text{red}}} \Phi^q s^q, \quad (12)$$

while conforming to the constraints (13), (14), and (15), which are described in detail below. It was already noted in Section 3.1 that there exists a wide array of meaningful weight functions Φ (and their continuous counter-parts φ). The above choice for Φ , the prey values, ensure that the ILP-solver picks for all operation points the data point with the least possible fuel consumption among all feasible choices.

The stack constraint

The solution $\widetilde{\text{SOL}}$ of our ILP should contain at most one element from each stack S_{ft} . This condition is reflected in the *stack constraint*

$$s_{ft} + \sum_{d^q \in S_{ft}} s^q = 1 \quad \text{for all } (f, t) \in [k] \times [k], \quad (13)$$

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 nonzero s_{ft} are the placeholders we mentioned above. They will be used in (15) to assign the penalty values to stacks that contribute no data point to the ILP-solution $\widetilde{\text{SOL}}$.

Formalizing the drivability constraint

To avoid engine damage, the discrete drivability constraint (5) ensures, that the variation speeds of all m actuators are constrained individually by nonnegative constants Δ_a for $a \in [m]$. To write this condition as a constraint in an ILP, let $d^q \in S_{f_t}$, and d^r be a data point in either neighboring stack $S_{f_{t\pm 1}}$ or $S_{f_{t\pm 1}}$. Then the *drivability constraints* (5) can be expressed as

$$s^q + s^r \leq 1 \quad \text{if} \quad |[d^q]_a - [d^r]_a| \geq \Delta_a \quad \text{for any } a \in [m]. \quad (14)$$

Note that this is a secant constraint, since the data points compared are contained in neighboring stacks.

Formalizing the emission constraint

The emission condition (7) 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

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

In the following, the weights ω_{f_t} represent the mean resistance time of the rectangles OP_{f_t} in the given test cycle. For simplicity of notation, the mean resistance time (cf. Section 3.2) on the rectangle corresponding to the stack S^q containing d^q is denoted by ω^q . Then the *emission constraint* can be formulated as follows:

$$\sum_{q=1}^{\delta_{\text{red}}} \omega^q [d^q]_p s^q + \sum_{(f,t) \in [k] \times [k]} \omega_{f_t} \bar{e}_p s_{f_t} \leq e_p \quad \text{for all } p \in E. \quad (15)$$

The second summand of the left hand side of the inequality serves as the above-mentioned penalty term which compensates for stacks that contribute no data point to the ILP-solution $\widetilde{\text{SOL}}$.

Remark 2 We want to determine the size of our ILP. There are k^2 equalities arising from the stack constraints. Further, it has $m \cdot 2 \cdot k \cdot (k - 1)$ drivability constraints and $|E|$ linear emission inequalities. There are k^2 stack decision variables and δ_{red} decision variables s^q , one for each data point in \widehat{F}_{red} . In the situation of the simulation data presented below, where $k = 16$, and $m = 8$, this results in 256 equalities and $3840 + 8$ inequalities which constrain the ILP-solution $\widetilde{\text{SOL}}$. Moreover, there are 256 stack decision variables. The size of \widehat{F}_{red} naturally varies throughout the calibration process, and may range between 10.000 and 100.000.

4.6 Closing the Gaps in $\widetilde{\text{SOL}}$

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_{ad} which is high-dimensional, as is the range of F . Naturally, this approach cannot produce a sufficient coverage of U_{ad} . In particular, often several stacks S_{f_t} of the k -operation field will contain no data point that contributes to the solution $\widetilde{\text{SOL}}$ of the integer linear program.

For each such non-contributing stack S_{ft} we proceed as follows. First, we construct a local model of F as follows. If S_{ft} lies in the interior of the operation field, we pick $N = m(m+1)/2$ data points

$$\{(u^{q1}, y^{q1}), (u^{q2}, y^{q2}), \dots, (u^{qN}, y^{qN})\}$$

from neighboring stacks and compute a Newton type polynomial that interpolates the points $\{u^{q1}, u^{q2}, \dots, u^{qN}\}$ as done in the local fit step IV. If an empty stack S_{ft} lies on the boundary of the operation field, then the local model is computed by calculating secants of neighboring data points in the nearest stacks and extending them linearly.

Second, we find, e.g., via Newton's method, some point $\tilde{u} \in U_{ad}$ such that under our local model of F we have $(\tilde{u}_{freq}, \tilde{y}_{torq}) \in S_{ft}$. In a ball about \tilde{u} we perform randomized measurements, e.g., using a normal distribution centered at \tilde{u} , until we find a point u with $(u_{freq}, y_{torq}) \in S_{ft}$ and which satisfies the drivability constraints. Due to the continuity of the manifold map there must exist a neighborhood of such points. The data point $d_{ft} := (u, y)$ is then added to the solution as the representative of the stack S_{ft} . Our way of picking the d_{ft} ensures that the completed solution map conforms to the drivability constraint and we thus have $SOL \in \Omega_k$.

For pollutant p the penalty value is \bar{e}_p , the upper bound of the interval $J_p = [\underline{e}_p, \bar{e}_p]$ of noncritical values. Hence, any recorded value of pollutant p , by which the penalty value is replaced, is smaller than or equal to \bar{e}_p . As a consequence, the total emission of a complete solution map SOL cannot exceed that of \bar{SOL} and must thus conform to the emission constraint if the preliminary solution does. Since we also have $SOL \in \Omega_k$, as noted above, the so-completed solution map does indeed solve problem (8).

5 AVL Engine Model

For our experiments in Section 6, we replace the test-bench with a 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 [2, p. 69ff] and [21].

5.1 Actuators of the AVL Engine Model

Revolution frequency of the crankshaft (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 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 the main-injection. 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.

Table 2 Actuator-intervals during various calibration runs. Δ^{\max} denotes the maximal variation speed of the actuator in respective units per second. LT/HT = low/high torque, FR = free variation of torque; cf. Sections 6.1 and 6.2.

Act.	Unit	Δ^{\max}	Basic, LT	Basic, HT	Basic, FR	Full Calibration
RF	$\frac{1}{\text{min}}$	10	1000–2600	1000–2600	1000–2600	1000–2600
IF	$\frac{\text{mm}^3}{\text{cycle}}$	0.1	6–10	50–60	6–60	6–60
RP	hPa	100	295677 / 405677	295677	295677	295677–1126537
AF	$\frac{\text{mg}}{\text{stroke}}$	5	300	300	300	275–991
TG	int	1	30	30	30	30–85
MT	$^{\circ}\text{CA}$	0.2	0 / 10	0	0	0–10
PI	$\frac{\text{mm}^3}{\text{cycle}}$	0	1	1	1	1
PT	μs	10	1540	1540	1540	1540–2565

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. It stands out among the other measurands since it provides the second coordinate axis of the operation field.

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 open. 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

As explained in Section 4, we subdivide the engine optimization into two phases. During the first phase, which we call *basic calibration*, measurements are taken for a prescribed amount of time in regions which are known to be critical to any calibration effort, regardless of the specific engine. The basic calibration is concluded by passing the obtained data to Algorithm 1, which is run without the emission constraint (15) in the ILP. If constraint (15) is omitted, there always exists a feasible solution $\widetilde{\text{SOL}}$ to the ILP, which may contain several gaps, though, due to the drivability constraint and empty stacks. Hence, the while loop exits immediately and the gaps in $\widetilde{\text{SOL}}$ are closed.

In the second phase, which we call *full calibration*, a solution map conforming to all constraints is obtained by building on and refining the preliminary solution map of the first phase. During full calibration, Algorithm 1 runs automatically and without time limit until it returns a complete solution map SOL.

One could, in theory, omit the first phase and turn the calibration procedure into a fully automatic one by letting all actuators range freely on their whole domain right from the start.

However, carefully applying engineering-knowledge in the first phase by bounding some actuators, fixing others, and ignoring the very restrictive emission constraints results in the quick gathering of a meaningful base set of data which then merely has to be refined in the sequel. In our experience, this approach considerably reduces the number of measurements required.

Throughout this section the precision parameter k is set to 16, and this yields 256 representative data points for the solution map SOL. This determines the shapes of Figures 5ff.

Remark 3 In the test bench scenario, measurements are taken, though not necessarily stored, in regular time intervals. A realistic frequency is one measurement per second. We take this frequency as the basis of our translation of the number of measurements in the simulation into real-world time. Below, we state the costs of our method in real-world time, as all computation times occurring during the different steps of Algorithm 1 are negligible in comparison to the days, or even weeks, that the physical experiments on a test bench can take.

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. Here a *run* means 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 preset 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 Table 2.

A good foundation for the optimization process 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. During the first run we measure the low torque region by limiting the IF-interval to 6–10 mm³/cycle, while the static actuators are set to values that support the creation of low torque operation points; cf. Table 2. This run takes 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. We measure the low torque region twice because it represents, in particular, the situation during startup, where low engine temperatures lead to unclean combustion, which causes high HC and CO emissions; cf. for example [16]. Hence, a detailed image of the engine behavior in this area of the operation field is desirable.

Afterwards, the IF-interval is reset to 50–60 mm³/cycle, so that the high torque regions can be measured. The static actuators are set to values that support high torque operation points; cf. Table 2. Again, the run takes 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; cf. Table 2. This region is measured for 6 hours, too. After 24 hours, we get a coarse picture of the engine behavior.

As indicated above, the so-obtained data are used as a base set for a first run of Algorithm 1, albeit with a deactivated emission constraint. Due to the omission of constraint (15), the while loop exits immediately as a feasible solution SOL to the ILP can always be found in this case. Due to empty stacks and the drivability constraints, which are still active, there may still be gaps in SOL, though. These are closed subsequently by interpolation-guided

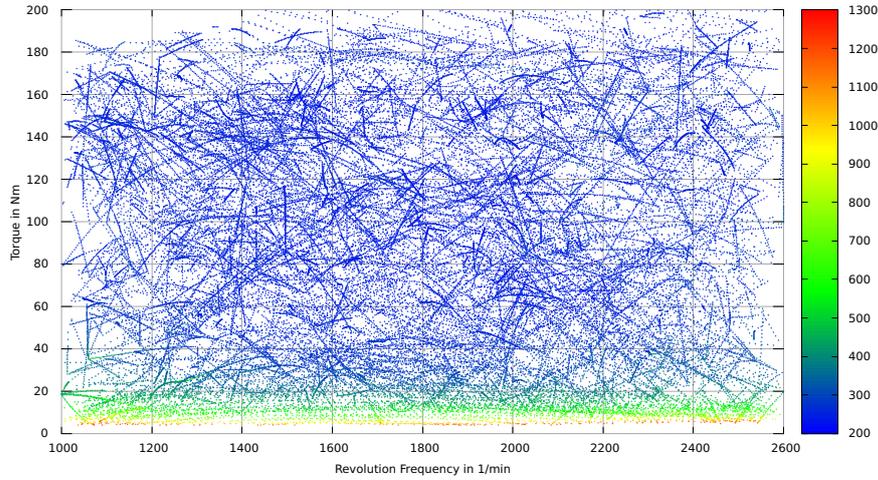


Figure 4 Operation field saturation after basic calibration, before closing of gaps.

measurements; cf. Section 4.6. For the comparison of an engine map, i.e., a single component of a solution map, that respects the drivability constraint to one that does not, see Figure 5.

It took an additional 45.8 hours to close all measurement holes. Figure 4 depicts the saturation of the operation field before this completion. The average fuel consumption for 100 kilometers of the so-derived preliminary solution is 4.65 liters in the NEDC. 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.

6.2 Full Calibration

During the full calibration phase the only static actuator is pilot injection. All others range over their whole respective domains. With these presets, Algorithm 1 runs automatically and without predefined time limit until it returns a solution map SOL.

The optimization is performed, first, for the NEDC, then for the RANDOM cycle. An NEDC solution map that conforms to the EURO 4 norm is obtained after 68.51 hours, one that conforms to EURO 5 after 88.41 hours. It turns out that the NEDC EURO 5 solution map conforms to all EURO 4 constraints with respect to the RANDOM cycle. That is, it is also a EURO 4 solution map for the latter. A EURO 5 solution map 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 map of equivalent quality, i.e., a solution map conforming to all constraints given by the different EURO norms, the uniform grid approach consistently required 15 to 20 times as many measurements, whereby we again mean performed

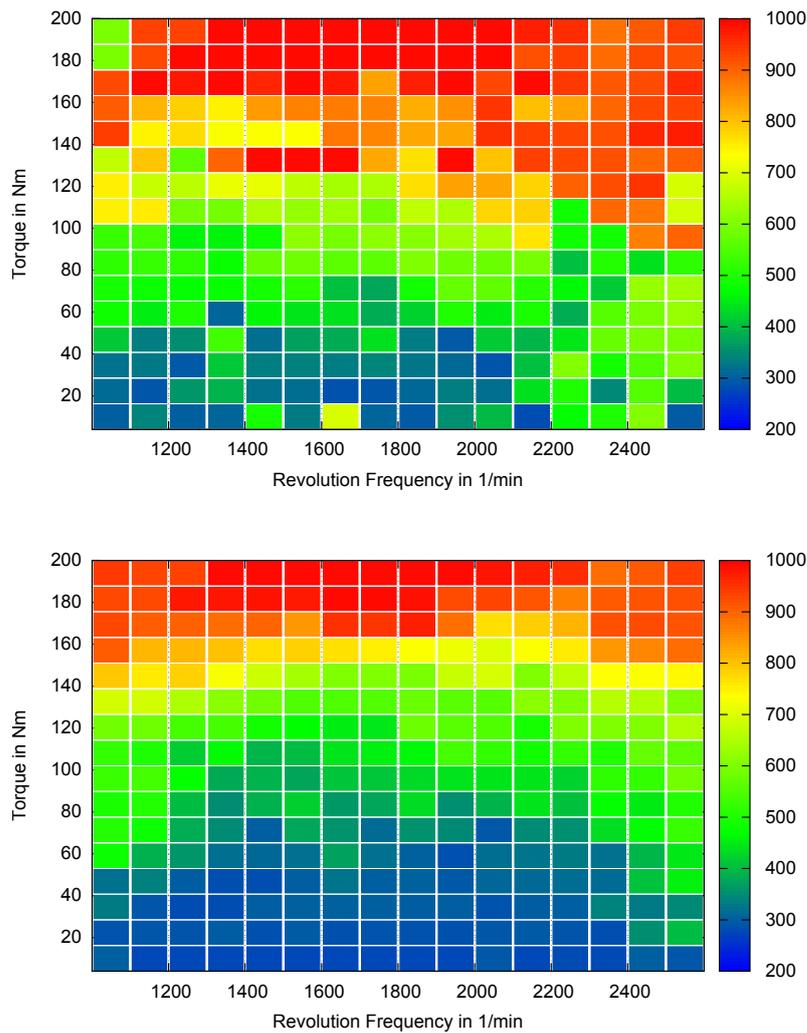


Figure 5 Top: Solution for actuator AF without drivability constraint. Bottom: AF settings of EURO 5 solution respecting the drivability constraint.

measurements, not stored ones. This translates into a real-world measurement time of weeks instead of days for a solution map of comparable quality.

Remark 4 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 map 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

Figure 6 displays the lower NO_x output on the better part of the operation field for the RANDOM calibration in comparison to a calibration for the NEDC.

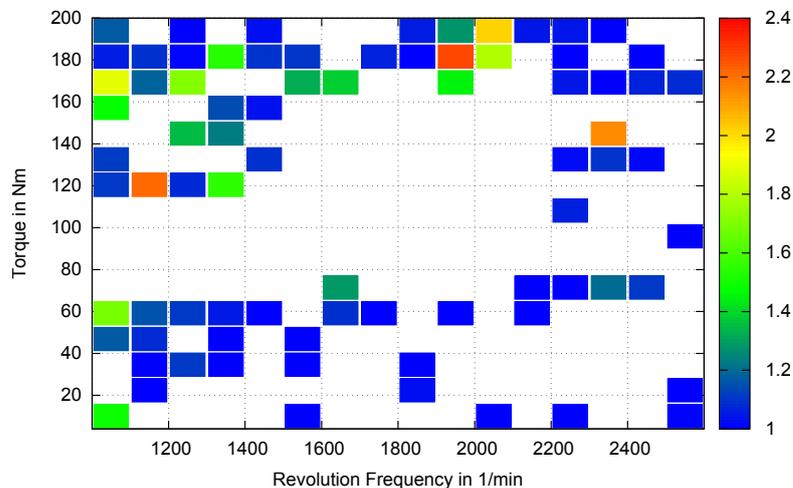


Figure 6 Operation points where calibration for RANDOM cycle yields higher NO_x emission than in NEDC. Colors represent the ratio of the emission values.

Figures 7 and 8 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 of it. In Figure 9 one can observe that the solution map 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 a solution for 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 strongly nonlinear or even 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.

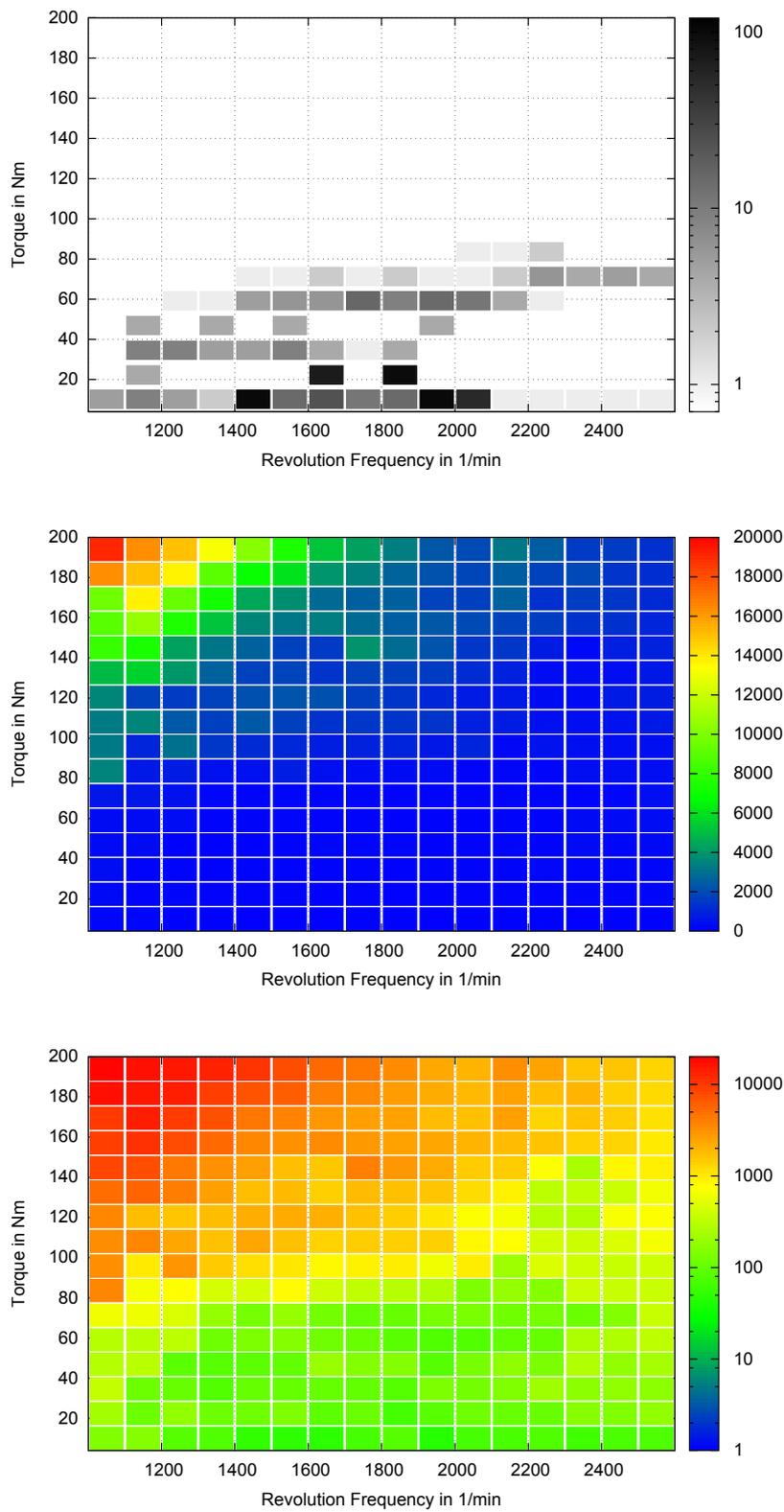


Figure 7 Top: Weighting table of the NEDC. Middle: Resulting NO_x emissions (in g/h) after calibration for the NEDC. Bottom: NO_x emissions with logarithmic scale.

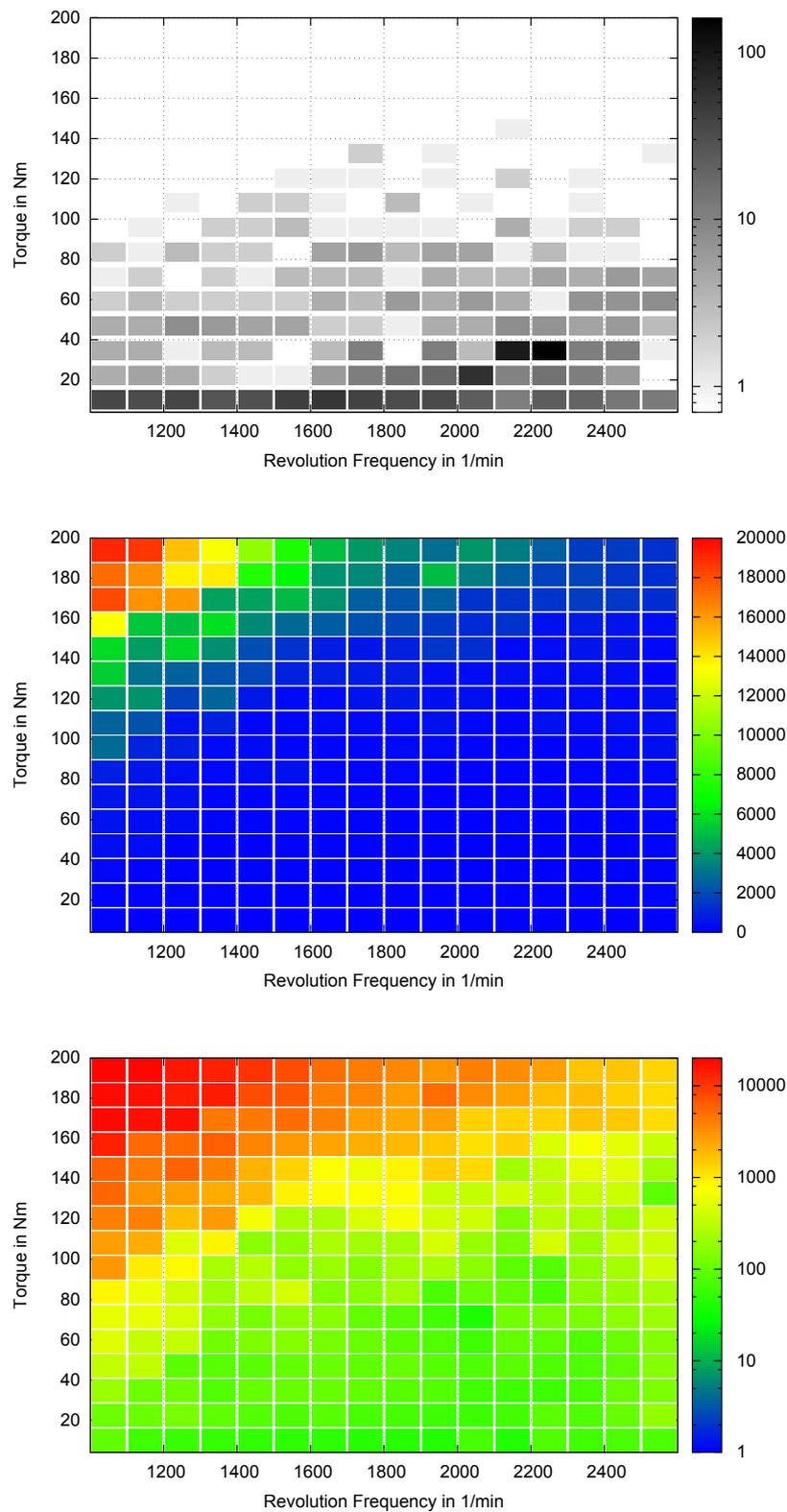


Figure 8 Top: Weighting table of the RANDOM cycle. Middle: Resulting NO_x emissions (in g/h) after calibration for the RANDOM cycle. Bottom: NO_x emissions on a logarithmic scale.

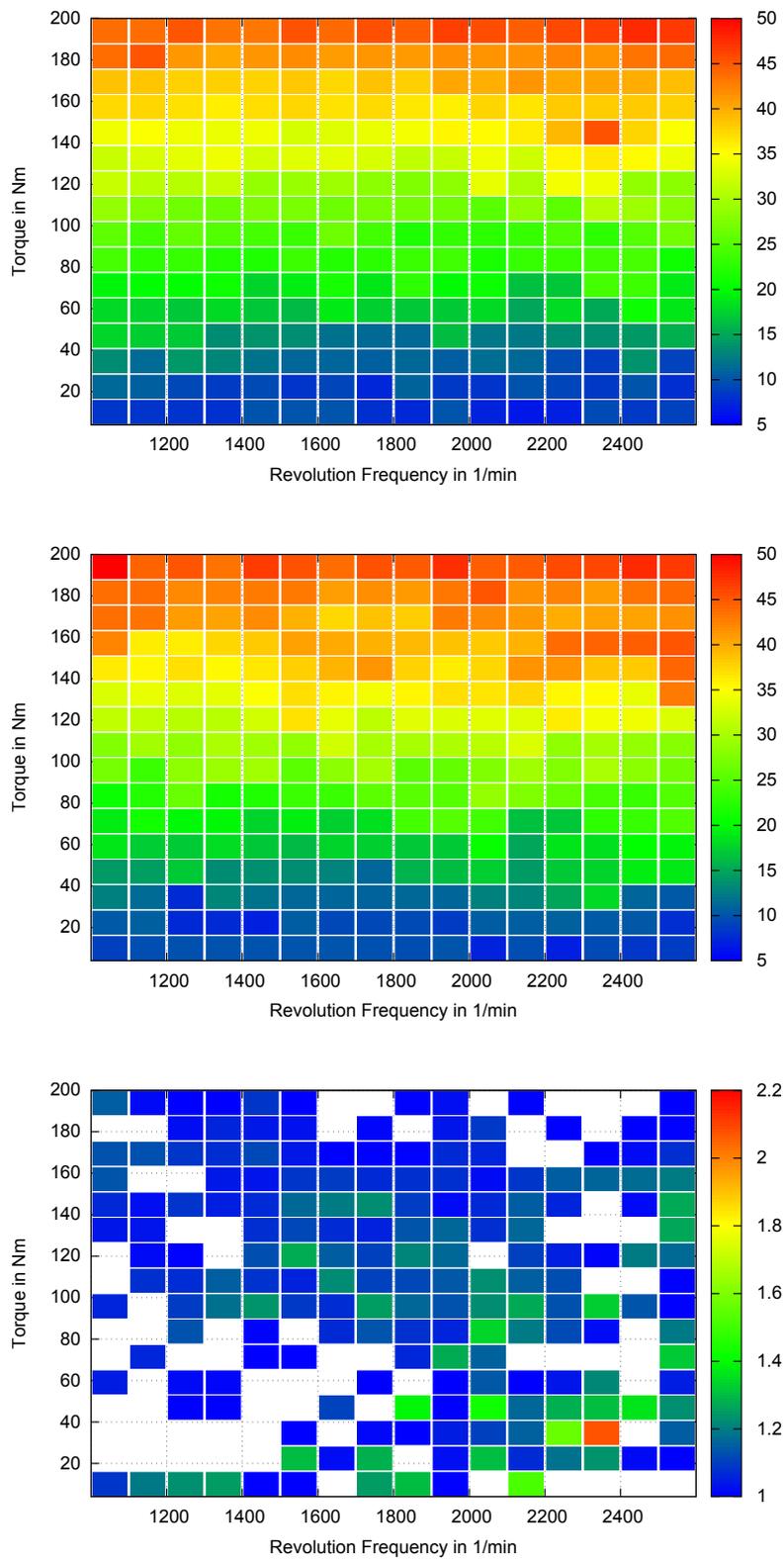


Figure 9 Actuator IF after calibration for the NEDC (top) and RANDOM cycle (middle). Bottom: Operation points with higher values in calibration for RANDOM cycle than for NEDC. Colors represent ratio of consumption values.

The output of the algorithm is a solution map SOL 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 in improved values for pollutant emission and fuel consumption near strongly nonlinear or even 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 (from weeks down to days) in comparison to uniform grids, without a loss of overall quality. Moreover, the resulting solution maps 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 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.

Acknowledgements We thank two anonymous reviewers for their comments that helped to improve the presentation of the paper.

References

1. Andoni, A., Indyk, P.: Nearest neighbors in high-dimensional spaces. In: C.D. Tóth, J.E. Goodman, J. O'Rourke (eds.) *Handbook of Discrete and Computational Geometry*, chap. 43. CRC Press (2018). 3rd edition
2. Burggraf, T.: Development of an automatic, multidimensional, multicriterial optimization algorithm for the calibration of internal combustion engines. Ph.D. thesis, TU Darmstadt (2015)
3. Cook, W.: *Combinatorial optimization*. Wiley, New York (1998)
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*, Third Edition, 3rd edn. The MIT Press (2009)
5. FEZ: Topexpert (2004). <http://www.fev.com/en/what-we-do/software-and-testing-solutions/products/testing/calibration/topexpert-suite>
6. Gschweidl, K., Pfluegl, H., Fortuna, T., Leithgoeb, R.: Steigerung der Effizienz in der modellbasierten Motoren-Applikation durch die neue CAMEO Online DoE-Toolbox. *ATZ – Automobiltechnische Zeitschrift* **103**(7), 636–643 (2001). DOI 10.1007/BF03226227
7. Isermann, R.: *Modellgestützte Steuerung, Regelung und Diagnose von Verbrennungsmotoren*. Springer, Berlin Heidelberg (2003). DOI 10.1007/978-3-642-55698-2
8. Isermann, R.: *Elektronisches Management motorischer Fahrzeugantriebe*. ATZ/MTZ-Fachbuch. Vieweg+Teubner, Wiesbaden (2010). DOI 10.1007/978-3-8348-9389-5
9. Isermann, R.: *Engine Modeling and Control*. Springer, Berlin Heidelberg (2014). DOI 10.1007/978-3-642-39934-3
10. Knödler, K., Poland, J., Fleischhauer, T., Mitterer, A., Ullmann, S., Zell, A.: Modellbasierte Online-Optimierung moderner Verbrennungsmotoren. *MTZ – Motortechnische Zeitschrift* **64**(6), 520–526 (2003). DOI 10.1007/BF03227107
11. Kruse, T., Kurz, S., Lang, T.: Modern statistical modeling and evolutionary optimization methods for the broad use in ECU calibration. *IFAC Proceedings Volumes* **43**(7), 739–743 (2010). DOI 10.3182/20100712-3-DE-2013.00031. 6th IFAC Symposium on Advances in Automotive Control
12. Martini, E., Voß, H., Töpfer, S., Isermann, R.: Efficient engine calibration with local linear neural networks. *MTZ worldwide* **64**(5), 19–22 (2003). DOI 10.1007/BF03227584
13. MathWorks: Model-based calibration toolbox (2018). <https://de.mathworks.com/products/mbc.html>
14. Mayeres, I., et al.: Transitions towards a more sustainable mobility system. Tech. Rep. EEA Report No. 34/2016, European Environment Agency (2016). <https://www.eea.europa.eu/publications/term-report-2016>
15. Merker, G.P., Schwarz, C., Teichmann, R.: *Grundlagen Verbrennungsmotoren*. Springer, Berlin Heidelberg (2018). DOI 10.1007/978-3-658-19212-9

16. Merkisz, J., Pielecha, J.: Particulate matter emissions during engine start-up. In: Nanoparticle Emissions From Combustion Engines, pp. 47–60. Springer, Cham (2015). DOI 10.1007/978-3-319-15928-7_5
17. Scholtes, S.: Introduction to piecewise differentiable equations. Springer (2012)
18. Schrijver, A.: Theory of linear and integer programming. John Wiley & Sons (1998)
19. Schüler, M., Hafner, M., Isermann, R.: Einsatz schneller neuronaler Netze zur modellbasierten Optimierung von Verbrennungsmotoren. MTZ - Motortechnische Zeitschrift **61**(10), 704–711 (2000). DOI 10.1007/BF03227308
20. Sierksma, G.: Linear and Integer Programming: Theory and Practice, 2nd edn. CRC Press (2001)
21. Vogels, M., et al.: DoE Model “Compression ignition engine”. AVL LIST GMBH, Graz (2013)