An Algorithm for Selecting Measurements with High Information Content Regarding Parameter Identification

Christian Potthast[0000−0001−6963−8254]

Robert Bosch GmbH, Corporate Sector Research and Advance Engineering
Robert-Bosch-Campus 1, 71272 Renningen, Germany
christian.potthast@de.bosch.com

Abstract. Reducing the measurement effort that is made for identification of parameters is an important task in some fields of technology. This work focuses on calibration of functions running on the electronic control unit (ECU), where measurements are the main expense factor. An algorithm for information content analysis of recorded measurement data is introduced that places the calibration engineer in the position to shorten future test runs. The analysis is based upon parameter sensitivities and utilizes the Fisher-information matrix to determine the value of certain measurement portions with respect to parameter identification. By means of a simple DC motor model the algorithm's working principle is illustrated. The first use on a real ECU function achieves a measurement time reduction of 67 % while a second use case opens up new features for the calibration of connected cars.

Keywords: Parameter identification · Fisher-information matrix · Local sensitivity · Measurement information · Measurement period reduction.

1 Introduction

The software of an electronic control unit (ECU) in a passenger car is an extensive program consisting of a couple of self-contained functions with strict interfaces and a vast number of parameters. These functions are designed to run in real-time and to model complex physical behaviour with a rough model structure only. But nevertheless a high flexibility regarding the output quantities is achieved at the same time since much data is stored in lookup tables. Most of the parameters for ECU functions result from these 1D- or 2D-lookup tables.

Since ECU models strongly rely on data tables, adapted product-specific parameter calibration has to be done again and again for each vehicle type making this process even more time-consuming and costly. By far the most expensive task in calibration are the measurements, i.e. driving on public roads or on test areas, and in the majority of cases more measurement time is carried out and recorded than would actually be necessary.

In practice the tuning process of the parameter values is either done manually or by means of an optimization algorithm. This work is mostly related

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

to cases where an optimizer is used since manual calibration requires a more rigid measurement schedule with long holding times at fixed operating points and hence leaving lesser leeway for compression. Investigations of the optimization algorithm itself are not subject of this article because prior studies have shown that a gradient-based optimizer for nonlinear least-squares curve fitting problems is doing well on the special class of ECU functions.

This paper presents an algorithm that analyses existent measurements piecewise and separates important sections providing new information from sections without content of further value. The main aim is to give advice on how to shorten similar test runs in the future. Secondly, the algorithm may indicate possibilities on how to speed up the parameter optimization. However, this only works out if measurement parts can be left out for simulation.

The basic technology is the well-known Fisher-information matrix, which has been used intensively for the design of experiments [1], [2] in various scientific fields. The Fisher matrix is also used for prioritization of parameters in order to identify most sensitive parameters first [3], [4] and hereby supporting a more target-oriented estimation process. However, the design of test runs from scratch by utilizing typical eigenvalue-based criteria is impractical for isolated ECU functions: a real test run produces input signals for the function that cannot be defined prior to the real experiment because of influences from the environment, the test track or road traffic.

The following section states the theoretical background before the algorithm is described in section 3 by means of a simple DC motor model. The successful application is demonstrated in section 4 for a new calibration feature of the Connected Car and for a real ECU function, where a drastic time reduction of 67 % is achieved.

2 Theoretical Background

For this paper a nonlinear dynamic model in state space formulation is assumed

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}), \quad t > 0, \quad \boldsymbol{x}(0) = \boldsymbol{x_0} \\ \boldsymbol{y}(t) &= \boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\theta}), \quad t \geq 0, \end{aligned} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of states, $\boldsymbol{y} \in \mathbb{R}^m$ is the vector of model outputs, $\boldsymbol{u} \in \mathbb{R}^q$ is the vector of model inputs and $\boldsymbol{\theta} \in \mathbb{R}^L$ is the parameter vector. The initial conditions for the states are $\boldsymbol{x_0} \in \mathbb{R}^n$.

Calibration aims at finding the unknown parameters $\boldsymbol{\theta}$, which result in the best fit between model outputs $\boldsymbol{y}(t, \boldsymbol{\theta})$ and measured responses $\boldsymbol{y}^M(t)$ from the real process under consideration. The error $\boldsymbol{e}(t, \boldsymbol{\theta})$ denotes the difference between measured and simulated outputs:

$$\boldsymbol{e}(t, \boldsymbol{\theta}) = \boldsymbol{y}^M(t) - \boldsymbol{y}(t, \boldsymbol{\theta}). \tag{2}$$

The root mean square error (RMSE) is typically used to quantify the deviation of a simulated signal consisting of N instants of time obtained with parameter

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

set $\boldsymbol{\theta}$ from a measurement of the i-th quantity of interest within the output vector $\boldsymbol{y}$:

$$RMSE(\boldsymbol{\theta}) = \sqrt{\frac{\sum_{k=1}^{N} e_i^2(t_k, \boldsymbol{\theta})}{N}}. \tag{3}$$

2.1 The Error Stochastics

Assuming that real measurements include all kinds of systematic errors as well as noise from the sensor, the measurement $\boldsymbol{y}^M$ is considered to have a deterministic and a stochastic part:

$$\boldsymbol{y}^M(t) = \boldsymbol{y}^{M,det}(t) + \boldsymbol{\epsilon}(t). \tag{4}$$

Putting Eq. (4) into Eq. (2) gives:

$$\boldsymbol{e}(t, \boldsymbol{\theta}) = \boldsymbol{y}^{M,det}(t) - \boldsymbol{y}(t, \boldsymbol{\theta}) + \boldsymbol{\epsilon}(t) = \boldsymbol{\eta}(t, \boldsymbol{\theta}) + \boldsymbol{\epsilon}(t). \tag{5}$$

Thus, the total error consists of a deterministic part $\boldsymbol{\eta}$ and a stochastic part $\boldsymbol{\epsilon}$. The stochastic part $\boldsymbol{\epsilon}$ is assumed to be Gaussian white noise, where the samples are independent of each other and have a normal distribution with zero mean:

$$\epsilon_i(t_k) \sim \mathcal{N}\left(0, \sigma_{\epsilon_i}^2\right) \quad \text{for } i = 1, \ldots, m \quad \text{and} \quad k = 1, \ldots, N. \tag{6}$$

The probability density function $p(\epsilon_i(t_k))$ for one time sample t_k from the i-th output signal therefore is:

$$p(\epsilon_i(t_k)) = \frac{1}{\sqrt{2\pi\sigma_{\epsilon_i}^2}} \exp\left(-\frac{\epsilon_i^2(t_k)}{\sigma_{\epsilon_i}^2}\right). \tag{7}$$

For one time sample t_k and multiple output signals ($m > 1$) it holds:

$$p(\boldsymbol{\epsilon}(t_k)) = \prod_{i=1}^{m} p(\epsilon_i(t_k)) = (2\pi)^{-\frac{m}{2}} \left(\det \boldsymbol{C}_{\boldsymbol{\epsilon}}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{m} \frac{\epsilon_i^2(t_k)}{\sigma_{\epsilon_i}^2}\right), \tag{8}$$

where $\boldsymbol{C}_{\boldsymbol{\epsilon}} \in \mathbb{R}^{m \times m}$ is the diagonal covariance matrix of the independent measurement error

$$\boldsymbol{C}_{\boldsymbol{\epsilon}} = \mathrm{E}\left[(\boldsymbol{\epsilon} - \mathrm{E}(\boldsymbol{\epsilon})) \cdot (\boldsymbol{\epsilon} - \mathrm{E}(\boldsymbol{\epsilon}))^{\top}\right] = \mathrm{E}\left[\boldsymbol{\epsilon} \cdot \boldsymbol{\epsilon}^{\top}\right] = \begin{pmatrix} \sigma_{\epsilon_1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{\epsilon_2}^2 & & \\ \vdots & & \ddots & \\ 0 & & & \sigma_{\epsilon_m}^2 \end{pmatrix}. \tag{9}$$

For multiple time samples ($N > 1$) and multiple output signals ($m > 1$) it holds:

$$p(\boldsymbol{\epsilon}) = \prod_{k=1}^{N} p(\boldsymbol{\epsilon}(t_k)) = (2\pi)^{-\frac{m \cdot N}{2}} \prod_{k=1}^{N} \left(\det \boldsymbol{C}_{\boldsymbol{\epsilon}}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\sum_{k=1}^{N}\sum_{i=1}^{m} \frac{\epsilon_i^2(t_k)}{\sigma_{\epsilon_i}^2}\right). \tag{10}$$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

In parameter estimation the exact (or true) values $\boldsymbol{\theta}^*$ of the parameter vector $\boldsymbol{\theta}$ are unknown. If during estimation the iteratively determined parameter values are close to the true values and under the assumption that the model represents all systematic properties of the process, i.e.

$$\boldsymbol{\theta} \to \boldsymbol{\theta}^* : \ \boldsymbol{\eta}(t, \boldsymbol{\theta}) = 0 \ \Rightarrow \ \boldsymbol{\epsilon}(t) = \boldsymbol{y}^M(t) - \boldsymbol{y}(t, \boldsymbol{\theta}^*) \tag{11}$$

the density function for $\boldsymbol{\theta} \to \boldsymbol{\theta}^*$ can be approximated in the following form:

$$p(\boldsymbol{\theta}) = (2\pi)^{-\frac{m \cdot N}{2}} \prod_{k=1}^{N} (\det \boldsymbol{C}_{\boldsymbol{\epsilon}})^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \sum_{k=1}^{N} \sum_{i=1}^{m} \frac{\left(\boldsymbol{y}^M(t_k) - \boldsymbol{y}(t_k, \boldsymbol{\theta})\right)^2}{\sigma_{\epsilon_i}^2} \right). \tag{12}$$

2.2 The Fisher-Information Matrix

The general formula for the Fisher-information matrix at a parameter vector $\hat{\boldsymbol{\theta}}$ is derived from the probability density function and is as follows:

$$\boldsymbol{\mathcal{I}}(\hat{\boldsymbol{\theta}}) = \mathrm{E}\left[\left. \frac{\partial \log p(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\hat{\boldsymbol{\theta}}} \cdot \left. \frac{\partial \log p(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right|_{\hat{\boldsymbol{\theta}}}^{\top} \right]. \tag{13}$$

Due to the log function the gradient from Eq. (12) has the compact form:

$$\begin{aligned} \frac{\partial \log p(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \sum_{k=1}^{N} \sum_{i=1}^{m} \frac{\boldsymbol{y}^M(t_k) - \boldsymbol{y}(t_k, \boldsymbol{\theta})}{\sigma_{\epsilon_i}^2} \cdot \frac{\partial \, \boldsymbol{y}(t_k, \boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}} \\ &= \sum_{k=1}^{N} \left(\frac{\partial \, \boldsymbol{y}(t_k, \boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}} \right)^{\top} \cdot \boldsymbol{C}_{\boldsymbol{\epsilon}}^{-1} \cdot \left(\boldsymbol{y}^M(t_k) - \boldsymbol{y}(t_k, \boldsymbol{\theta}) \right). \end{aligned} \tag{14}$$

Some further calculations and simplifying assumptions stated in [3] lead to

$$\begin{aligned} \boldsymbol{\mathcal{I}}(\hat{\boldsymbol{\theta}}) &= \sum_{k=1}^{N} \left(\left(\frac{\partial \, \boldsymbol{y}(t_k, \boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}} \right)^{\top} \Bigg|_{\hat{\boldsymbol{\theta}}} \cdot \boldsymbol{C}_{\boldsymbol{\epsilon}}^{-1} \cdot \left(\frac{\partial \, \boldsymbol{y}(t_k, \boldsymbol{\theta})}{\partial \, \boldsymbol{\theta}} \right) \Bigg|_{\hat{\boldsymbol{\theta}}} \right) \\ &= \sum_{k=1}^{N} \left(\boldsymbol{S}_{\boldsymbol{y}}^{\top} \big|_{t_k, \hat{\boldsymbol{\theta}}} \cdot \boldsymbol{C}_{\boldsymbol{\epsilon}}^{-1} \cdot \boldsymbol{S}_{\boldsymbol{y}} \big|_{t_k, \hat{\boldsymbol{\theta}}} \right). \end{aligned} \tag{15}$$

The Fisher-information matrix $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{L \times L}$ is a symmetric positive semidefinite matrix, which can be calculated easily as summation over all instants of time. The only necessary quantities are the constant covariance matrix of measurement noise $\boldsymbol{C}_{\boldsymbol{\epsilon}}$ and the time-variant output sensitivity matrix $\boldsymbol{S}_{\boldsymbol{y}}$. Both are briefly described in sections 2.4 and 2.5.

In order to achieve comparability between parameters often the normalized Fisher matrix $\boldsymbol{\mathcal{I}}_{\boldsymbol{n}}(\boldsymbol{\theta})$ is used. Since parameter values may differ by several orders of magnitude, $\boldsymbol{\mathcal{I}}(\boldsymbol{\theta})$ is multiplied from left and right with a diagonal matrix with parameter values on its main diagonal

$$\boldsymbol{\mathcal{I}}_{\boldsymbol{n}}(\boldsymbol{\theta}) = \boldsymbol{\theta}^{\top} \boldsymbol{I}_{\boldsymbol{L}} \, \boldsymbol{\mathcal{I}}(\boldsymbol{\theta}) \boldsymbol{I}_{\boldsymbol{L}} \, \boldsymbol{\theta}, \tag{16}$$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

where $\boldsymbol{I_L}$ is the $L \times L$ identity matrix. It is worth mentioning that parameter values of zero pose a problem to the described normalization because the corresponding normalized sensitivity is set to zero and erroneous zero rows and columns appear in $\boldsymbol{\mathcal{I}_n}$. The problem may be handled by setting zero parameters to small values instead or, if reasonable, by renouncing normalization at all. Although the selection algorithm introduced in section 3 is based upon the normalized Fisher matrix this is not absolutely necessary since the algorithm uses a parameter individual assessment, where the comparability between parameters is virtually unnecessary.

2.3 Parameter Variances

The Cramér-Rao inequality

$$\boldsymbol{C_\theta} \geq \boldsymbol{\mathcal{I}}^{-1}(\boldsymbol{\theta^*}) \tag{17}$$

says that the inverse Fisher-information matrix is a lower bound for the covariance matrix $\boldsymbol{C_\theta}$ of the parameter estimation error [5], [6], [7], where $\boldsymbol{C_\theta}$ is defined by:

$$\boldsymbol{C_\theta} = \mathrm{E}\left[(\boldsymbol{\theta} - \boldsymbol{\theta^*}) \cdot (\boldsymbol{\theta} - \boldsymbol{\theta^*})^\top\right]. \tag{18}$$

The algorithm for evaluation of information content of measurements introduced in section 3 uses the square roots of the main diagonal elements of $\boldsymbol{\mathcal{I}}$ (or $\boldsymbol{\mathcal{I}_n}$ respectively), i.e. the standard deviations for individual parameters j

$$\sigma_{\theta_j} = \sqrt{\mathcal{I}^{-1}_{jj}(\boldsymbol{\theta^*})}, \quad j = 1, \ldots, L. \tag{19}$$

2.4 Covariance of the Measurement

There are several reasonable possibilities to determine the covariance of the uncoupled measurement error $\boldsymbol{C_\epsilon}$ used in Eq. (15). Three are specifyed below:

a) Stationary Measurement Phase If the measurement contains stationary periods of time the variance can simply be determined as sample variance

$$\sigma^2_{\epsilon_i} = \frac{1}{N_{stat}} \sum_{k=1}^{N_{stat}} \left(y^M_{i,stat}(t_k) - \bar{y}^M_{i,stat}\right)^2 \tag{20}$$

from the N_{stat} measurement samples assumed to be stationary or as in [8]: with $N_{stat} - 1$ in the denominator of Eq. (20). The constant value $\bar{y}^M_{i,stat}$ is the mean value of the N_{stat} measured samples under consideration.

b) Difference between Measurement and Simulation In dynamic measurements it is often impossible to find time periods, which can be regarded as stationary. In these cases a formula proposed in [9] may be used:

$$\sigma^2_{\epsilon_i} = \frac{1}{N - L} \sum_{k=1}^{N} \left(y^M_i(t_k) - y_i(t_k, \boldsymbol{\theta})\right)^2. \tag{21}$$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

c) Data Sheet The third possibility is to exploit given or known information about the applied sensor, e.g. from the sensor manufacturer's data sheet or just from experiential knowledge of experts.

2.5 Sensitivities

It should be pointed out that in this context the sensitivities are the local output sensitivities and should not be mixed up with the global sensitivities, whose meaning and method of calculation is completely different [10]. Basically there are three different options for determining the local output sensitivities:

a) Sensitivity Differential Equation System Defining shorter forms of state and output sensitivity as

$$\boldsymbol{S_x} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}} \ , \quad \boldsymbol{S_y} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\theta}} \tag{22}$$

both can be calculated using the sensitivity differential equation system (SDES)

$$\begin{aligned} \dot{\boldsymbol{S}}_{\boldsymbol{x}} &= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \boldsymbol{S_x} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} \ , \quad t > 0 \ , \quad \boldsymbol{S_x}(0) = \frac{\partial \boldsymbol{x_0}}{\partial \boldsymbol{\theta}} \\ \boldsymbol{S_y} &= \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}} \boldsymbol{S_x} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} \ , \quad t \geq 0. \end{aligned} \tag{23}$$

Eq. system (23) is usually created with a computer algebra system using symbolic differentiation and solved numerically afterwards.

The advantage of solving the SDES over other methods like the simple difference quotient (see following paragraph) is that the resulting sensitivities are very precise, if the numerical integration is handled well. The main drawback of this kind of calculation is that the model must be able to be formulated as continuous ODE system as stated in Eq. (1). For many practical models this is already a criterion of exclusion, because models of real processes in industry often contain switching parts like lookup tables or other types of switching operations that require case-by-case analysis. Furthermore solving of the SDES together with the model equations can be computationally expensive due to a large number of state variables.

b) Difference Quotient If model equations are not accessible directly or switching parts are included, often the simple difference quotient, which is basically equivalent to an external numerical differentiation, is used. The sensitivity vector $\boldsymbol{s_j}$, i.e. the derivative of all outputs to a single parameter θ_j, arises from:

$$\boldsymbol{s_j}(t) = \frac{\partial \boldsymbol{y}(t, \boldsymbol{\theta})}{\partial \theta_j} \approx \frac{\boldsymbol{y}(t, \boldsymbol{\theta} + \Delta\theta_j \boldsymbol{e_j}) - \boldsymbol{y}(t, \boldsymbol{\theta})}{\Delta\theta_j}, \quad j = 1, \ldots, L, \tag{24}$$

where $\Delta\theta_j$ is the deflection of the j-th parameter that can be positive (forward) or negative (backward). $\boldsymbol{e_j} \in \mathbb{R}^L$ is the j-th unit vector. The major drawback

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

of this procedure is the accuracy that decreases with larger absolute deflections $|\Delta\theta_j|$. On the other hand very small deflections lead to catastrophic cancellation [11]. In addition to simple forward and backward differences also methods of higher order (two-sided and fourth-order differences) are common. A guiding value for choosing $\Delta\theta_j$ depending on machine precision, reference point and method order is given in [12].

c) Automatic Differentiation Another important variant for calculating sensitivities is automatic differentiation (AD). Assuming the model is available as computer code (in the first instance C-Code) the idea is to create a second code that calculates the necessary derivates by exploiting the chain rule additionally to the model equations itself, see [13], [14] and many others.

3 The Selection Algorithm

The purpose of the selection algorithm is to find measurements or just measurement parts with high information content with respect to a subsequent parameter optimization. A reverse interpretation, i.e. finding the least informative parts, is also reasonable and intended. Since the Fisher matrix is summed up over time and can be interpreted as measure for information content, every time sample increases information. Information increase in turn is reflected in parameter variance decrease, which is used as measure for importance. For practical reasons measurements are analysed section-wise instead of sample-wise from start to end. Based on a threshold value the proposed algorithm decides from section to section whether it is important or not. Algorithm 1 gives the selection process as pseudocode.

Since the decision about section importance is made immediately and together with past sections the algorithm is suitable and explicitly designed for online usage. However, for the sake of clarity Algorithm 1 demonstrates the offline use case. The first step (line 1) is partitioning the measurement in N_s sections. It is worth mentioning that these sections may, but need not be of equal length. The first section acts as initialization for a global Fisher matrix representing the collected information of all important sections (line 2). The outer for-loop (line 5) runs over all sections and combines the new information with the already selected one (line 7). Inside the inner for-loop, that runs over all parameters, a section is chosen as important if at least one parameter related standard deviation decreases by at least the predefined percentage threshold value δ_{thr} (line 10). A positive decision for any parameter updates the global information (line 11), while a negative decision dismisses the combined information without global update.

Before a continued and summarizing discussion of assumptions and limits of the proposed algorithm is given in section 3.2, the following section 3.1 demonstrates algorithm as well as results with the help of a simple DC motor model.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

Algorithm 1 Calculate set of important sections $\mathcal{S}$

```
Split measurement(s) in N_s sections
I^global ← calculate Fisher-information matrix for section 1 acc. to Eq. (15)
σ_θ^global ← main diagonal from inverted I^global (Eq. 19)
S ← {1}    // set section 1 as first element of set of important sections
for i = 2 to N_s do
    I^local ← calculate Fisher-information matrix for section i acc. to Eq. (15)
    I^comb. ← I^global + I^local
    σ_θ^comb. ← main diagonal from inverted I^comb. (Eq. 19)
    for j = 1 to L do
        if σ_θj^comb. ≤ σ_θj^global · (1 − δ_thr/100) then
            I^global ← I^comb. ; σ_θ^global ← σ_θ^comb.
            S ← S ∪ {i}    // add section i to set of important sections
            break
        end if
    end for
end for
return S
```

3.1 Exemplary DC motor model

The exemplary motor model is an adapted version from [15] with external load torque and mechanical transmission. The model equations are as follows:

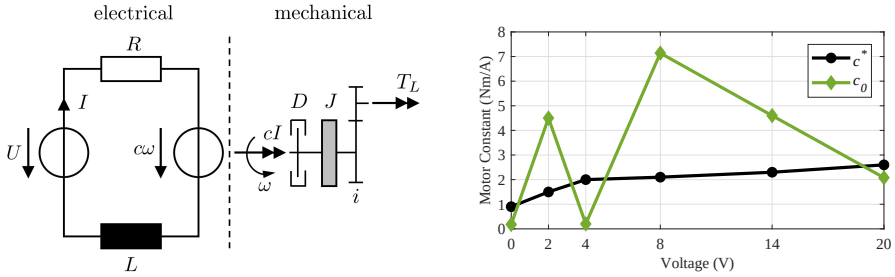
$$\begin{aligned} \dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{I}(t) \\ \dot{\omega}(t) \end{bmatrix} &= \begin{bmatrix} -R/L & -c/L \\ c/J & -D/J \end{bmatrix} \cdot \begin{bmatrix} I(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} 1/L & 0 \\ 0 & -1/(i \cdot J) \end{bmatrix} \cdot \begin{bmatrix} U(t) \\ T_L(t) \end{bmatrix} \\ \boldsymbol{y}(t) = \begin{bmatrix} I(t) \\ \omega_m(t) \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 30/\pi \end{bmatrix} \cdot \begin{bmatrix} I(t) \\ \omega(t) \end{bmatrix}. \end{aligned} \tag{25}$$

The equivalent circuit is shown in Fig. 1(a). Model inputs are voltage U and load torque T_L, outputs are armature current I and angular velocity ω_m in the non-SI unit rpm. All parameters (R, L, D, J, i, c) are assumed to be unknown. In order to design the model being more realistic in terms of the target applications (ECU functions) the motor constant is modelled as 1D-lookup table $c = c(U)$, i.e. the actual value of c depends on the input voltage and has to be interpolated between supporting points as shown in Fig. 1(b).

Assuming that there is no need for the 1D-lookup table to fulfil a particular shape requirement (e.g. monotony), it is transformed into six independent parameters $c_1, \ldots, c_6$ in place of c and hence extending the length of parameter vector $\boldsymbol{\theta}$ from 6 to 11.

Fig. 2 shows the preconceived inputs as well as the output measurements generated synthetically by a simulation with true parameters $\boldsymbol{\theta}^*$ (black line) with superimposed white noise in accordance with Eq. (4). The green lines show simulation results obtained with parameter start vector $\boldsymbol{\theta_0}$. Amongst others Table 1 lists the values of $\boldsymbol{\theta_0}$ and $\boldsymbol{\theta}^*$.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41



(a) Equivalent circuit. (b) Lookup table for motor constant c.

Fig. 1. Input and output signals for the DC motor model.

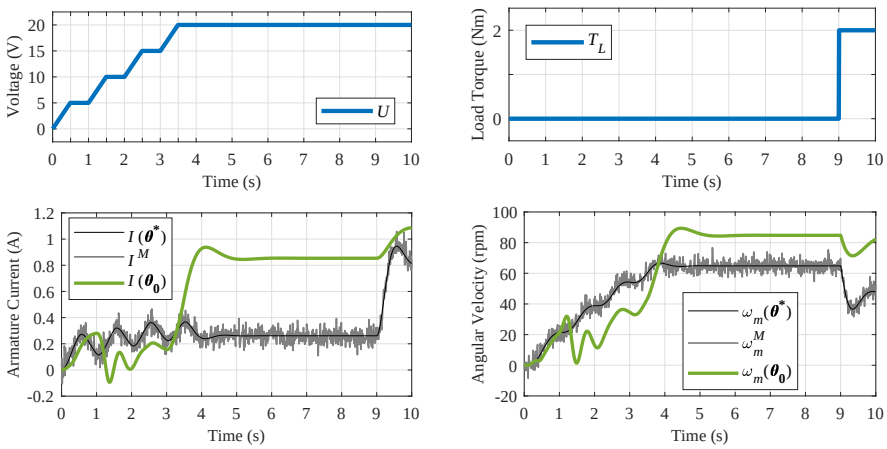


Fig. 2. Input and output signals for the DC motor model.

Since the model outputs have different ranges of values, the normalized root mean square error (NRMSE), see Eq. (26), is used as scalar dimensionless error measure for both outputs together.

$$NRMSE(\boldsymbol{\theta}) = \sqrt{\frac{\sum_{k=1}^{N} \frac{\left(I^M(t_k) - I(t_k,\boldsymbol{\theta})\right)^2}{\Delta I^2} + \sum_{k=1}^{N} \frac{\left(\omega_m^M(t_k) - \omega_m(t_k,\boldsymbol{\theta})\right)^2}{\Delta \omega_m^2}}{2N}} \tag{26}$$

In contrast to Eq. (3) the residuals are normalized with the corresponding ranges of values ΔI and $\Delta\omega_m$ that are obtained from the deterministic measurement parts $I(t,\boldsymbol{\theta}^*)$ and $\omega_m(t,\boldsymbol{\theta}^*)$:

$$\begin{aligned} \Delta I &= \Delta I(\boldsymbol{\theta}^*) = \max\left(I(t,\boldsymbol{\theta}^*)\right) - \min\left(I(t,\boldsymbol{\theta}^*)\right) \\ \Delta\omega_m &= \Delta\omega_m(\boldsymbol{\theta}^*) = \max\left(\omega_m(t,\boldsymbol{\theta}^*)\right) - \min\left(\omega_m(t,\boldsymbol{\theta}^*)\right). \end{aligned} \tag{27}$$

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

Table 1. Different predefined / optimized parameter sets with resulting NRMSE values

Symbol	R	L	D	J	i	c_1	c_2	c_3	c_4	c_5	c_6	$NRMSE$
Unit	Ω	H	kg/(m^{2}s)	kg m^2	–	Nm/A						–
$\boldsymbol{\theta^*}$	9.0	2.0	0.1	0.1	1.2	0.9	1.5	2.0	2.1	2.3	2.6	0.0499
$\boldsymbol{\theta_0}$	1.8	7.0	0.2	0.05	4.2	0.18	4.5	0.2	7.14	4.6	2.08	0.5760
$\bar{\boldsymbol{\theta}}_{\{1,\dots,37\}}$	8.67	2.05	0.10	0.10	1.18	0.00	2.20	1.99	2.13	2.30	2.62	0.0497
$\bar{\boldsymbol{\theta}}_{\mathcal{S}}$	8.13	2.01	0.10	0.10	1.32	0.00	2.35	1.96	2.15	2.33	2.63	0.0504
$\bar{\boldsymbol{\theta}}_{\mathcal{R}1}$	7.60	2.22	0.10	0.09	4.20	7.93	1.70	2.01	2.18	2.35	2.67	0.1155
$\bar{\boldsymbol{\theta}}_{\mathcal{R}2}$	8.39	2.04	0.10	0.11	1.19	0.29	2.42	2.04	2.35	2.04	2.63	0.0550
$\bar{\boldsymbol{\theta}}_{\mathcal{R}3}$	9.84	1.99	0.10	0.09	4.20	5.54	26.91	2.07	2.05	2.21	2.59	0.1194

As it is recorded in Table 1 the true parameters $\boldsymbol{\theta^*}$ lead to a NRMSE of 0.0499 which pretty much reflects the predetermined standard deviations that originally have been put into the noise distributions for the measurement generation, i.e. $\mathcal{N}_I\left(0,(0.05\cdot\Delta I)^2\right)$ for the current output and $\mathcal{N}_{\omega_m}\left(0,(0.05\cdot\Delta\omega_m)^2\right)$ for the angular speed output respectively. The largest NRMSE in Table 1 is unsurprisingly produced with the start parameters $\boldsymbol{\theta_0}$, which are intentionally designed to give a worse fit, while the lowest NRMSE is the result of an optimized vector $\bar{\boldsymbol{\theta}}_{\{1,\dots,37\}}$ based upon the complete measurement and hence can be regarded as reference. The index set $\{1,\dots,37\}$ indicates that the time samples from all 37 sections have been used for optimization, since the total measurement period of 10 s has been partitioned into 37 sections of nearly equal length, see visualization in Fig. 3 (left).

The relevant quantities for the selection algorithm are the standard deviations that are calculated according to Eq. (19) on the basis of the normalized Fisher matrix Eq. (16). The required covariance of measurement noise $\boldsymbol{C_\epsilon}$ is a diagonal matrix with the variances used in the already mentioned noise distributions $\mathcal{N}_I$ and $\mathcal{N}_{\omega_m}$ on the main diagonal. The required sensitivities are calculated using difference quotient formula Eq. (24) since the tabled motor constant c complicates the creation of the SDES, see Eq. system (23), drastically. The threshold value δ_{thr} is set to 60 %.

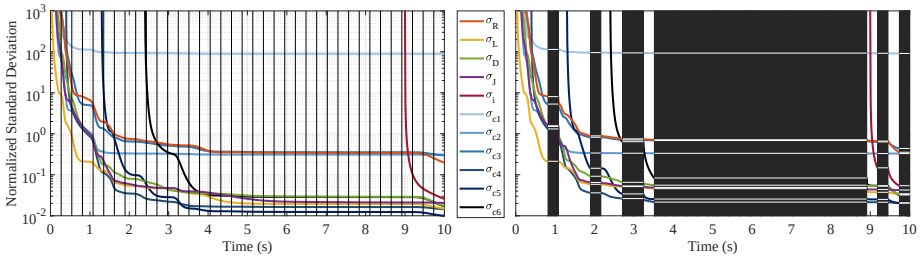


Fig. 3. Normalized standard deviations if the complete measurement is considered (left) and if only the most important sections (set $\mathcal{S}$) are considered (right).

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

Fig. 3 depicts the resulting parameter variations: on the left every instant of time is used for calculation while on the right it is assumed that the selection algorithm has been applied and each not selected section is skipped and highlighted black. As can be seen in Fig. 3 (right) the deviations are monotonically decreasing within each of the 11 sections in set $\mathcal{S}$ while remaining constant in the unpicked ones. It should be mentioned that all standard deviations start at ∞ and only accept real numbers when the corresponding parameter becomes sensitive for the first time.

The optimization result with all sections in $\mathcal{S}$ is given in Table 1 followed by three results for performance comparison purposes. The latter use section sets $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$ containing as many randomly selected sections as contained in $\mathcal{S}$. Of course, the given NRMSE values are calculated on the basis of the whole measurement and not only for the underlying measurement subset, i.e. the parameter set after optimization is used for a subsequent simulation of the whole 10 s and the result is put into Eq. (26). It turns out that the measurement parts in $\mathcal{S}$ produce a fit only slightly worse than the reference, while a random selection is less reliable performing sometimes comparably well ($\mathcal{R}_2$) and sometimes significantly worse ($\mathcal{R}_1$, $\mathcal{R}_3$).

3.2 Assumptions and Limitations

Some characteristics of the selection algorithm deserve further discussion:

- In view of measurement period reduction an unambitious partitioning generally leads to a couple of measurement snippets that are hard or impossible to combine into a realizable test run. Therefore a more anticipatory segmentation is essential for practical usage.
- In view of computation time reduction during optimization an adequate segmentation is also important since stopping the simulation and restarting at arbitrary points without having simulated the parts in between changes the characteristics of a dynamic model.
- The threshold value δ_{thr} is the main tuning parameter of the algorithm. It may vary between 0 % (select everything) and 100 % (select nothing except for section 1). The choice depends on the desired share of important segments. Values between 40 % and (more ambitious) 80 % have turned out to be sensible.
- The measurement section order is crucial for the resulting set $\mathcal{S}$. For online usage this is logical and unavoidable if an immediate decision is necessary. For offline usage it always has to be taken into account that the algorithm's decision is based upon the preceding measurements.
- Since the theory only holds for models without systematic errors, see Eq. (11), and true parameters, the used start parameter values should not be too far away from the true ones. Real applications in the context of ECU function calibration allow the use of start parameters from previous projects and therefore satisfy this precondition.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

– It would be unreasonable to expect that the algorithm always finds either the best sections or only the necessary ones. It must be kept in mind that the basis is just a stochastic analysis not necessarily valid for every single spot check. Also the fitting quality depends strongly on the last link in the chain: the optimization algorithm.

4 Real industry applications

Two applications show how the algorithm can be applied beneficially in industry.

4.1 Shortening of a Standard Measurement Program

The goal of the first application is to shorten an established measurement program used for calibration of an ECU function that calculates temperatures in the exhaust gas system of a passenger car. The task for the calibration engineer is to carry out the measurements and tune 59 parameters so that the model predicts measured temperatures as good as possible. The original program requires a car ride of several hours in total. For reasons of confidentiality more details about the model's interior structure and the measuring process cannot be revealed.

Although the function has two output signals (temperatures of exhaust gas and exhaust pipe), Fig. 4 only shows gas temperatures for the benefit of a more compact presentation. The upper digram in Fig. 4 shows the standard test run consisting of 11 independent real measurements separated by vertical lines. The measurement order does not matter, they are just plotted one after another in the same diagram in order to save space and to show the total length.

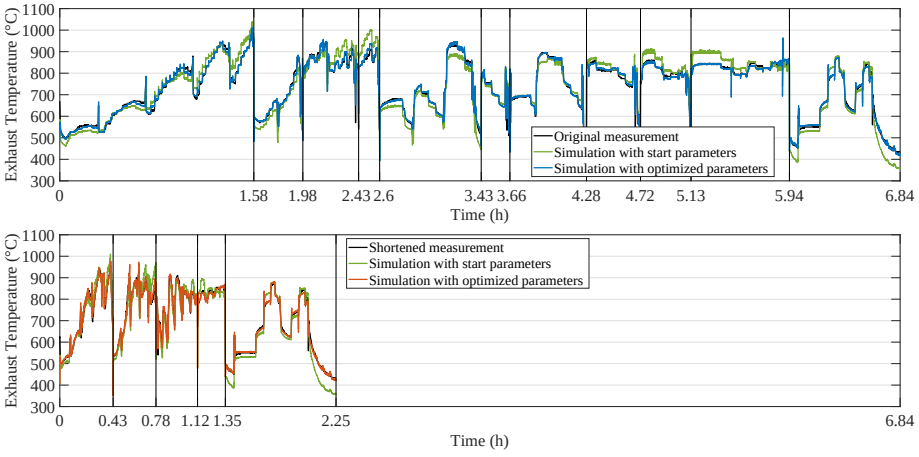


Fig. 4. Temperature characteristics of original and shortened measurement program.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

The segmentation for Algorithm 1 is done in such a way that some sections, which are necessary for preserving the conventional test run structure, are preselected. Using a threshold value $\delta_{thr} = 40\,\%$ the result from Algorithm 1 after some manual tuning leads to a shorter test run that was put into practice, see lower graph in Fig. 4.

Two optimizations of original and shortened test run result in different parameter sets reducing the initial error from approx. 60 °C to 11 °C in both cases. Applied to a third comparison measurement the two parameter sets produce similar good fits. The RMSE for parameters from the standard test is 11.78 °C and the parameters from the short test run lead to 12.67 °C. The slight worsening of almost 1 °C is very acceptable in view of the achieved time reduction of 67 % and the related cost saving.

4.2 Online Parametrization of Connected Cars

The second application deals with calibration data selection in the context of the Connected Car [16]. The use case is to decide online which parts of the car's recorded data is of value for calibration. The benefit from connecting cars in this setting is that parametrization can be carried out not only for a single car but for a fleet as well. Measurements can be transferred to a central cloud space or server and updated parameters can be sent back and flashed into the ECU software. The need for an importance decision of measurement data right in the car is caused by limitations of the data transmission via mobile net: the limiting factor is either bandwith (3G network) or cost (4G network).

The Connected Car application of this measurement selection algorithm is also subject of a related published patent application [17].

5 Conclusion

In this paper an algorithm was presented that distinguishes important measurement parts from unimportant ones. The algorithm may either be used online during the test run for immediate decisions as used e.g. for reducing measurement data being transferred via mobile network in the Connected Car application, see section 4.2. Or it may be used offline for analysing and shortening of pre-built measurement plans as shown for the exhaust temperature ECU function, see section 4.1.

Currently the simplest method for sensitivity calculation is used: the difference quotient. Especially in the context of ECU software it should be switched to automatic differentiation. Future work will concentrate on a modified version of the algorithm focussing on offline uses cases where the order in which the algorithm reads the measurement sections should no longer affect its selection result. The future calibration trend towards Big Data (i.e. measure and record everything) will increase the need for thinning out measurements and will open up further beneficial applications for the selection of important measurements parts.

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41

References

1. Atkinson, A. C.: The Usefulness of Optimum Experimental Designs. In: Journal of the Royal Statistical Society. Series B (Methodological) **58**(1), pp. 59-76 (1996)
2. Goodwin, G. C.: Identification: Experiment Design. In: System and Control Encyclopedia **4**, 2257-2264. Pergamon Press, Oxford (1987)
3. Majer, C. P.: Parameterschätzung, Versuchsplanung und Trajektorienoptimierung für verfahrenstechnische Prozesse. Fortschritt-Berichte Reihe 3, No. 538. VDI-Verlag Düsseldorf (1998)
4. Schmidt, A. P., Bitzer, M., Imre, Á. W., Guzzella, L.: Experiment-driven electrochemical modeling and systematic parameterization for a lithium-ion battery cell. In: Journal of Power Sources **195**, pp. 5071-5080 (2010). https://doi.org/10.1016/j.jpowsour.2010.02.029
5. Ljung, L.: System identification: Theory for the User. 2nd edn. Prentice Hall, Upper Saddle River (2006)
6. Norton, J. P.: An Introduction to Identification. Academic Press, London (1986)
7. Goodwin, G. C., Payne, R. L.: Dynamic System Identification: Experiment Design and Data Analysis, Vol. 136. Academic Press, New York (1977)
8. Zwillinger, D.: CRC Standard Mathematical Tables and Formulae. CRC Press, Boca Raton (1995)
9. Schittkowski, K.: Experimental Design Tools for Ordinary and Algebraic Differential Equations. In: Industrial & Engineering Chemistry Research **46**(26), pp. 9137-9147 (2007). https://doi.org/10.1021/ie0703742
10. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: Global Sensitivity Analysis. The Primer. John Wiley & Sons Ltd (2008)
11. Forsythe, G. E.: Pitfalls in Computation, or Why a Math Book isn't Enough. In: The American Mathematical Monthly **77**(9), pp. 931–956 (1970). https://doi.org/10.1080/00029890.1970.11992636
12. Schittkowski, K.: NLPQLP A Fortran Implementation of a Sequential Quadratic Programming Algorithm with Distributed and Non-Monotone Line Search – Users Guide, Version 4.2 (2014)
13. Griewank, A., Walther, A.: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. 2nd edn. Society for Industrial and Applied Mathematics (2008)
14. Rall, L. B., Corliss, G. F.: An Introduction to Automatic Differentiation. In: Computational Differentiation: Techniques Applications and Tools, pp. 1–18. SIAM, Philadelphia (1996)
15. Toliyat, H. A., Kliman, G. B.: Handbook of Electric Motors. 2nd edn. CRC Press, Boca Raton (2004)
16. Coppola, R., Morisio, M.: Connected Car: Technologies, Issues, Future Trends. In: ACM Computing Surveys (CSUR) **49**(3), pp. 46:1–46:36. ACM, New York (2016). https://doi.org/10.1145/2971482
17. Robert Bosch GmbH: Method and device for influencing a vehicle behavior. Inventors: Potthast, C., Bleile, T., Hagen, L., Petridis, K., Bitzer, M., Jarmolowitz, F. Filing date: 28 November 2016. Patent application no. WO 2017/093156 A1. Publication date: 8 June 2017

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22744-9_41