2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Automatic tuning of PID Controllers in Engine Control Units by means of Local Model Networks and Evolutionary Optimization

Christian H. Mayr, Nikolaus Euler-Rolle and Stefan Jakubek

***Abstract*— In this work a new approach for a fully automated calibration of nonlinear PID controllers and feedforward maps is introduced. Controller design poses a particularly challenging task in the application to internal combustion engines due to the nonlinear controller structure, which is usually prescribed by the manufacturer of the engine control unit (ECU). A dynamic local model network is used to represent the actual physical process as its architecture can beneficially be adopted for scheduling of the nonlinear controller parameters. The presented calibration technique uses a genetic algorithm to calibrate the nonlinear PID controller and a static model inversion to determine the feedforward map. Finally, an example demonstrates the effectiveness of the proposed method.**

I. Introduction

In engine control units (ECUs) usually discrete-time, nonlinear PID controllers with a specific structure are used for many control tasks such as the actuator position in a variable-turbine geometry (VTG) turbocharger for intake manifold pressure control. Basically, the controller gains are retrieved from nonlinear maps, which depend on engine load and speed. Further, there are additional parameters to distinguish between small and large control errors. Usually calibration engineers determine the parameters and maps within the ECU structure with testbed runs, test drives and a lot of expert knowledge. A model-based calibration method will help to increase the efficiency of the calibration workflow for ECUs. For this purpose it is reasonable to use local model networks (LMN), which approximate even strongly nonlinear dynamic processes by a network of locally linear dynamic submodels. Their approximation capabilities allow, or at least facilitate, the design of PID controllers for nonlinear systems. LMN from the family of multiple-model approaches (e.g. [1]) are a qualified approach because of their transparent structure and the possibility to incorporate prior (physical) knowledge, [2]. LMN interpolate between different local models, each valid in a certain operating regime. Each of these operating regimes represents a simple model, e.g. a linear regression model, describing the local dynamics.

This paper introduces a method for calibration of nonlinear PID controllers in ECUs using LMN. In this context two main tasks have to be solved:

- *Automatically determine feedforward maps*
  In ECUs there are two-dimensional feedforward maps, which usually depend on load and speed.
- *Automatically determine controller parameters*
  The gains of PID controllers (usually P, I, D and T_1 of the DT_1-Part) are two-dimensional maps, which, very much like the feedforward maps, usually depend on load and speed. In ECU PID control scheduling of the controller parameters is usually not only carried out along load and speed. In addition to these quite obvious scheduling variables, the control error itself is often used as a means for parameter scheduling. This approach is commonly understood as error signal adaption. It is noteworthy, that this introduces an additional nonlinearity into the closed-loop system.

The feedforward map is determined by a point-wise static inversion of the local model network. To determine the nonlinear PID maps a genetic algorithm (GA) is used which optimizes the closed loop performance of the control system. This performance is determined by simulating the closed-loop with proper input/reference signals. The performance is measured by the summation of the quadratic offset of the output from an expected output in each time-step. Well known model based, characteristics based or rule based methods for the autotuning of PID controllers [3], [4] mostly account for simple linear process models of low order only. Such methods could be applied to each local linear model in the LMN individually, but the overall performance of the closed-loop would remain unconsidered. To obtain good initial conditions for the optimization the local application of such an autotuning method is reasonable.

This paper is organized as follows: First, the nonlinear PID controller of ECUs is investigated in Section II. The architecture of local model networks is described in Section III. Section IV presents the methodology to design nonlinear ECU PID controllers. Next, in Section V the effectiveness of the proposed method is shown by means of an example, where the VTG position of a turbocharger is used for controlling the intake manifold pressure. Finally, the paper is concluded by some remarks in Section VI.

II. Architecture of PID Controllers used in ECUs

This section describes the architecture as well as the control algorithm of common PID controllers in ECUs. Figure 1 gives an overview of the architecture, which employs a PID controller with a DT_1-Part. The feedforward map, the controller gains as well as the reference map to determine the reference signal w, the actuating variable u_{fb} and the feedforward signal u_{ff} respectively, depend on engine load q (in mg/stroke) and speed n (in rpm). In contrast to the feedforward and the controller gain maps, the reference map

This work was supported by the Christian Doppler Research Association and AVL List GmbH

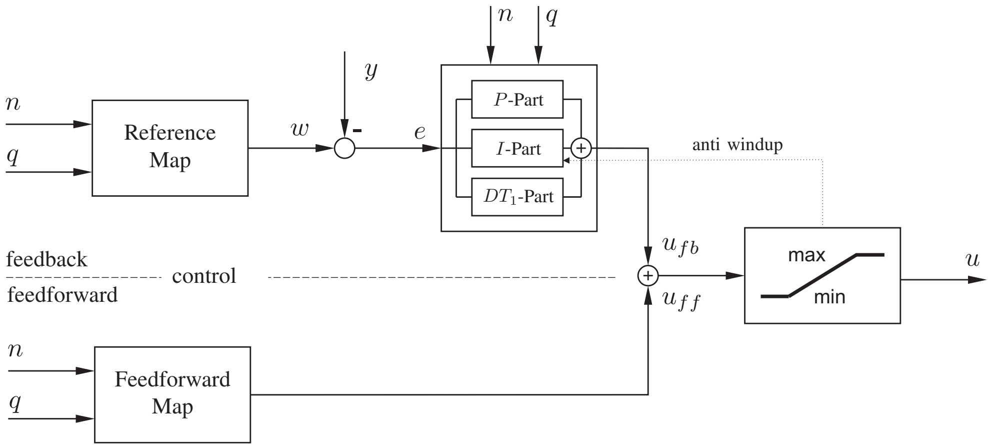

Fig. 1. Scheme of a nonlinear PID controller used in ECUs

is largely determined from emission limits beforehand and is usually prescribed for calibration engineers.

In addition to the above control architecture ECUs employ error signal adaption, which is applied to the control error e within each controller part individually and results in a nonlinearly modified error signal. For example, the adaption in the P-Part leading to the modified control error $\bar{e}_p$ is shown in Figure 2. All three gradients $K_{p,aPos}$, $K_{p,a0}$ and $K_{p,aNeg}$ as well as the two values $W_{p,pos}$ and $W_{p,neg}$ describing the small signal window are subject to optimization in the process of calibration.

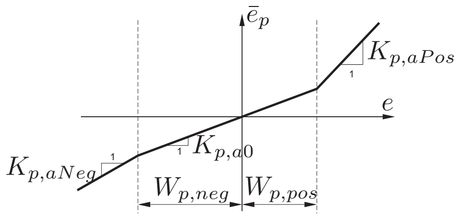

Fig. 2. Nonlinear characteristic of the error signal adaption within the P-Part

Figure 3 exemplarily shows the scheme for determining the actuating variable u_p within the P-Part of the nonlinear PID controller. Note that the dimensioned controller gain $K_{p,map}$ is multiplied with the adapted control error $\bar{e}_p$, which contains one or several of the constant dimensionless gradients $K_{p,a0}$, $K_{p,aPos}$ and $K_{p,aNeg}$. Thus, the resulting (time variant) gain depends on load q, speed n and the control error e itself.

All gains of the PID controller depend on maps, each of which depends on load and speed itself. The difference equations describing all parts of the nonlinear PID controller are given in the following, denoting the sampling time by T_s:

- P-Part
$$u_p(k) = K_{p,map}(n,q)\,\bar{e}_p(k) \tag{1}$$
with (2).
- I-Part
$$u_i(k) = u_i(k-1) + K_{i,map}(n,q)\,Q(e)\,e(k) \tag{3}$$

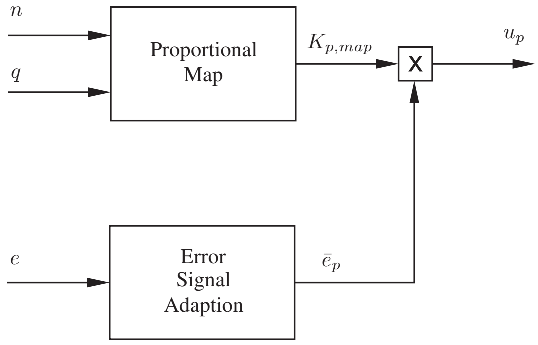

Fig. 3. Calculation scheme of the actuating variable u_p within the proportional part of the PID controller

with

$$Q(e) = \begin{cases} K_{i,aPos}\,T_s & e(k) > W_{i,pos} \\ K_{i,aNeg}\,T_s & e(k) < W_{i,neg} \\ K_{i,a0}\,T_s & \text{otherwise} \end{cases}$$

- DT_1-Part
$$u_d(k) = \exp\left(-\frac{T_s}{T_1}\right) u_d(k-1) + K_{d,map}(n,q)\,\bar{e}_d(k) \tag{4}$$
with (5).

As it can be seen in these equations, the error signal adaption of the P- and the DT_1-Parts results in nonlinear gain curves, whereas the error signal adaption of the I-Part is a simple parameter switch.

According to Figure 1, the P-, I- and DT_1-Parts are finally summed up to form the controller output

$$u_{fb}(k) = u_p(k) + u_i(k) + u_d(k) \tag{6}$$

As shown in this section, the architecture of the ECU PID controller is basically nonlinear and there are many parameters and maps, which influence the controller behavior. It is obvious, that it is complex to calibrate the ECU PID controller because of the high number of degrees of freedom.

$$\bar{e}_p(k) = \begin{cases} e(k)K_{p,aNeg} + (K_{p,a0} - K_{p,aNeg})W_{p,neg} & e(k) < W_{p,neg} \\ e(k)K_{p,aPos} + (K_{p,a0} - K_{p,aPos})W_{p,pos} & e(k) > W_{p,pos} \\ e(k)K_{p,a0} & \text{otherwise} \end{cases} \tag{2}$$

$$\bar{e}_d(k) = \begin{cases} \Delta e(k)K_{d,aNeg} + (K_{d,a0} - K_{d,aNeg})W_{d,neg} & \Delta e(k) < W_{d,neg} \\ \Delta e(k)K_{d,aPos} + (K_{d,a0} - K_{d,aPos})W_{d,pos} & \Delta e(k) > W_{d,pos} \\ \Delta e(k)K_{d,a0} & \text{otherwise} \end{cases} \tag{5}$$

with $\Delta e(k) = e(k) - e(k-1)$

III. Local Model Networks

In previous works LMNs have shown state-of-the-art approximation capabilities for internal combustion engines; e.g. [5], [6], [7] and the ability for PID controller design [8], [9]. Thus, in this work dynamic local model networks are used to approximate the nonlinear process to be controlled. In the context of combustion engines, a typical subsystem to be controlled is the intake manifold pressure for example.

As starting point for the description of LMNs, which follows in this section, their architecture is depicted in Figure 4.

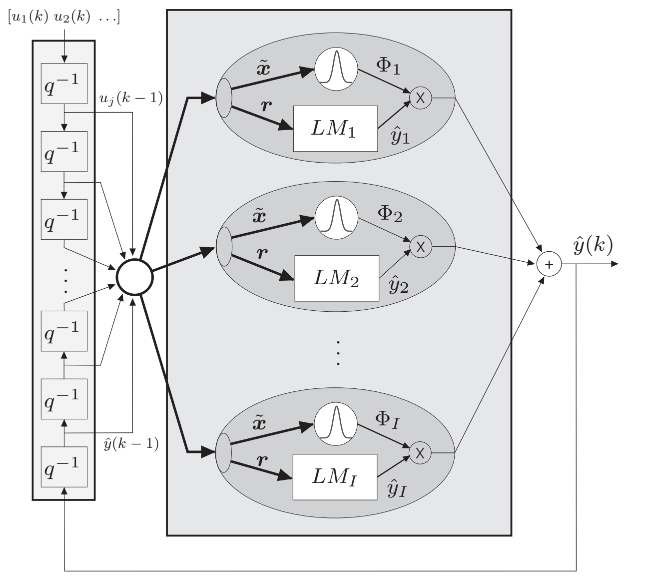

Fig. 4. Architecture of a local model network with external dynamics, [10]

The *local* model outputs

$$\hat{y}_i(k) = \boldsymbol{r}(k)\boldsymbol{\theta}_i, \quad \forall\mathcal{I} \tag{7}$$

where $\boldsymbol{r}(k)$ denotes the input vector for the rule consequents at time k and contains past inputs and outputs:

$$\boldsymbol{r}^T(k) = [u_i(k-1)\ \ldots\ u_I(k-m) \\ y(k-1)\ \ldots\ y(k-n)] \tag{8}$$

In (8) m and n denote the system order of the numerator and denominator respectively. A different choice of the input vector for the rule premises $\tilde{\boldsymbol{x}}(k)$ is advantageous because the mathematical complexity of the identification is reduced dramatically. All local estimations $\hat{y}_i(k)$ with the local parameter vectors

$$\boldsymbol{\theta}_i = \begin{bmatrix} b_1^{(i)} & \ldots & b_m^{(i)} & a_1^{(i)} & \ldots & a_n^{(i)} \end{bmatrix}^T \tag{9}$$

are used to form the global model output $\hat{y}(k)$ by weighted aggregation, see Figure 4:

$$\hat{y}(k) = \sum_{\mathcal{I}} \Phi_i(\tilde{\boldsymbol{x}}(k))\hat{y}_i(k). \tag{10}$$

Therein the validity functions are constrained to form a partition of unity:

$$\sum_{\mathcal{I}} \Phi_i = 1 \tag{11}$$

$$0 \le \Phi_i \le 1, \quad \forall\mathcal{I}. \tag{12}$$

This LMN uses recursive, axis-oblique partitioning strategy and smooth validity functions.

IV. Controller Calibration

This section describes the proposed methodology for automatic calibration of the nonlinear PID controller of ECUs. To recapitulate, it is required to determine the feedforward map, the maps of the PID controller parameters and the parameters of the error signal adaption as shown in Section II. The error signal adaption comprises 15 parameters overall: for each controller part three slope values ($K_{p,a0}$, $K_{p,aNeg}$, $K_{p,aPos}$, $K_{i,a0}$, $K_{i,aNeg}$, $K_{i,aPos}$, $K_{d,a0}$, $K_{d,aNeg}$, $K_{d,aPos}$) and two small signal window values ($W_{p,neg}$, $W_{p,pos}$, $W_{i,neg}$, $W_{i,pos}$, $W_{d,neg}$, $W_{d,pos}$).

Basically, each of the calibration tasks in Section IV-A, IV-B and IV-C is accomplished separately, although the previous result is considered by the following task. Subsequently, the methodology will be applied to control the intake manifold pressure by actuating a variable-geometry turbocharger of a light-duty internal combustion engine.

For the investigated case of controlling the intake manifold pressure by actuating a variable-geometry turbocharger, the maps of the ECU PID controller depend on load and speed. Thus, the trained such that control input u is the turbocharger actuator position, whereas load and speed are chosen as partition space dimensions

$$\mathbf{\Phi} = [n(k-1)\ q(k-1)]\,. \tag{13}$$

A. Determination of Feedforward Map

In current ECU structures static maps are a commonly accepted and easy to understand way of feedforward control. Dynamic LMN facilitate a direct way of steady state system inversion for the parametrisation of such maps. However, attention should be pointed to the fact, that LMN bear the potential of more sophisticated approaches of feedforward control such as dynamic system inversion of dynamic decoupling in the case of multi-input multi-ouput systems. In the presented application the feedforward map is generated by a point-wise static inversion of the open-loop state-space model by

$$u_{ff}(\mathbf{\Phi}) = w(\mathbf{\Phi})\frac{1-\sum_{j=1}^{n} a_j(\mathbf{\Phi})}{\sum_{j=1}^{m} b_j(\mathbf{\Phi})} \tag{14}$$

Therein, the reference signal $w(\mathbf{\Phi})$ depends on a given reference map according to Figure 1.

B. Determination of PID Maps

The calibration of the PID maps is based on an evolutionary optimization to optimize the *performance* of the closed-loop. For this purpose, a genetic algorithm ([11]) is used. It is noteworthy that the introduced concept needs not necessarily be used in combination with genetic algorithms. Other evolutionary algorithms such as Particle Swarm Optimization (PSO), [12], may also be used.
Due to the fact, that the proposed PID controller design method aims to optimize closed-loop performance the following performance criterion is considered:

Performance Criterion: The major task of the performance criterion is to quantify the quality of the PID controller in terms of its performance (e.g. rise time, overshoot...). For this purpose the performance criterion is based on given load and speed excitation signals. These signals cover the whole identified operating area of the LMN to capture the global nonlinear closed-loop performance. Ideally, load and speed signals should reflect as many realistic driving operations as possible in a preferably short time interval to keep the duration of its simulation limited. As the simulation has to be performed each time the criterion is evaluated, the length of this driving cycle strongly influences the overall computing time of the optimization.

To constitute synthetic signals, which are as realistic as possible, the load signal may exhibit jumps as the quantity of injection mass can be altered almost instantaneously in combustion engines. On the other hand, the gradient of the speed signal must be limited due to inertia. Usually the speed gradient is limited to 250 rpm/s.

By means of a given reference map, the reference signal $w(k)$ is obtained from the load and speed signals in each point in time. Figure 5 exemplarily shows excitation signals (top graph), a reference map for the intake manifold pressure as well as the resulting reference signal (bottom graph). As steps, which occur in the load signal, are directly reflected in the original reference signal $w(k)$, its application as ideal reference for optimization is not recommended. It would favor very fast controller setups with large overshoot and high actuating effort. Therefore an optimal output signal $y_{dmd}(k)$ is found by applying a moving average considering a few samples of $w(k)$.

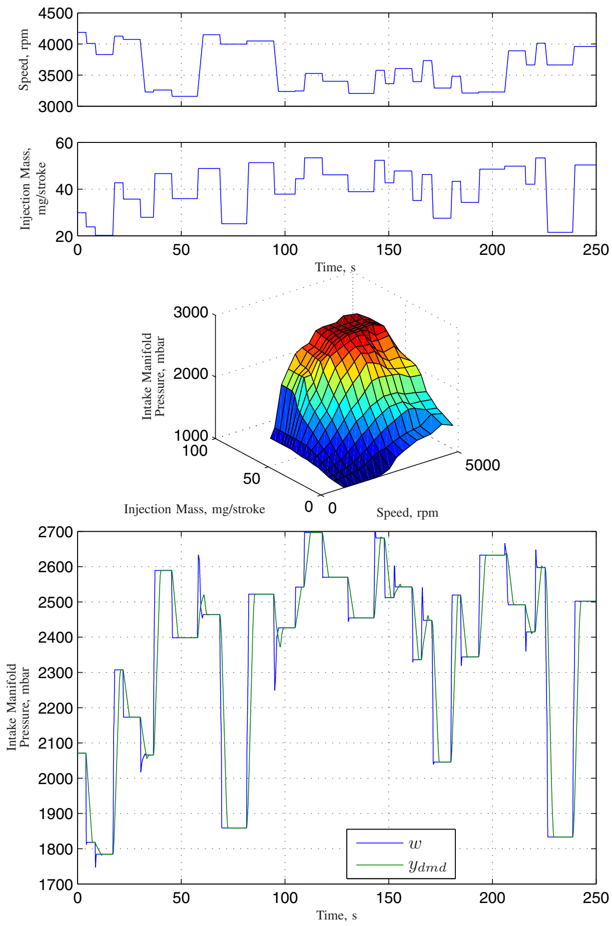

Fig. 5. In the upper panel: speed and load excitation signals, below: reference map for the intake manifold pressure for a light-duty Diesel engine, lower panel: resulting demanded output signals for the performance criterion.

In the evaluation of the performance criterion, this reference signal is applied in a closed-loop simulation for each genome. The fitness function compares the simulation output $y(k)$ to the demanded optimal output $y_{dmd}(k)$, which is derived from the reference signal $w(k)$. For a sequence of length K, the fitness function f_P for the performance assessment is the sum of squared errors

$$f_P = \sum_{k=1}^{K}(y_{dmd}(k) - y(k))^2. \tag{15}$$

C. Determination of the Error Signal Adaption

In the determination of the parameters describing the error signal adaption, the same approach as for the PID maps is chosen.

Again a GA with a criterion similar to that of Section IV-B is used. The PID maps, which have been found in the previous section, are now considered given and the GA is applied to the 15 parameters of the error signal adaption only. Slightly different speed and load signals are used to increase operation time in the large error windows, which is crucial to optimize these parameters. The main difference regarding the excitation lies in a larger gradient of the speed signal, which is increased to 500 rpm/s.

V. Example

In this section, an ECU calibration for actuating a variable-geometry turbocharger of a light-duty 4-cylinder Diesel engine to control the intake manifold pressure is investigated. The proposed method is compared to a conventional calibration, which has been achieved by a calibration engineer. It is represented by an in-vehicle measurement. Results of the proposed method have been found using simulations on a validated semi-physical engine model. As input data for the simulation the measurement data were used to ensure a fair comparison.

The following parameters have been applied in the identification of the dynamic local model network describing the intake manifold pressure:

- Engine Speed: $n = 3100 - 4400$ rpm
- Injection Mass: $q = 20 - 55$ mg/stroke
- Turbocharger actuator position: $u_{VTG} = 20\% - 96\%$
- 6 local models
- Input order: $m = 2$
- Output order: $n = 3$
- Partition space: $\mathbf{\Phi} = [n\ q]$ (as implemented in the ECU)

In the following, three calibrations will be compared:

- Conventional calibration (in-vehicle measurement)
- Two different calibration stages
  - *Initial Design*
    The feedforward map is determined as described in Section IV-A and local PID controller parameters are found by using the automated tuning feature `pidtune` of Matlab [4] applied to each local controller individually.
  - *Final Optimization*
    The feedforward map is used and PID controller parameters are found by the proposed method.

A comparison of intake manifold pressure in time domain is made in Figure 6. The simulated model output is compared to the training data. In most parts, sufficient dynamic approximation capabilities, but partial static offsets can be observed, which are caused by the restricted choice of engine speed and load in the partition space. By including other values into the partitioning, the model accuracy could be increased although such a choice would not be possible in the current ECU structure.

In Figure 7 simulation results for two stages of the controller calibration are depicted. The upper plot shows the intake manifold pressure simulated on a performance sequence (P_{dmd}) for the initial design (P_{ini}) and for the final calibration (P_{fin}). The bottom plot shows the cumulated absolute error $\sum_i |P_{dmd,i} - P_i|$ of the two calibration stages. As compared to the initial design, the cumulative error is reduced to 50% in the final design.

The final calibration has been achieved with a population size of 100 genomes and 50 generations.

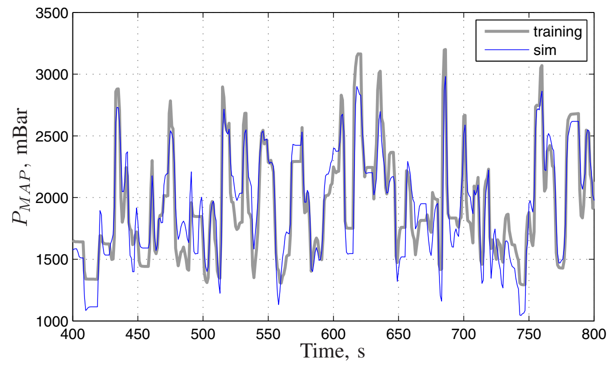

Fig. 6. Comparison of training data and its simulation by means of LMN

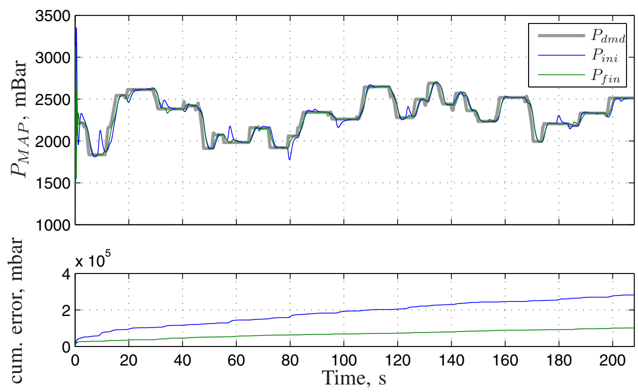

Fig. 7. Comparison of different controller calibration stages by means of the performance sequence cycle

The effectiveness of the proposed methodology and its advantages over conventional calibration are demonstrated by a test run using the conventional calibration. Manifold pressure and relevant input quantities such as engine speed or injection mass have been recorded by in-vehicle measurement over an interval of approximately 580 seconds. To give a general notion of how the test run looked like, Figure 8 gives a comparison of the measured reference performance (P_{meas}) with the two calibration stages being simulated using the measured inputs. It has to be said, that during the test run the in-vehicle controller has been deactivated during some periods with idle operation such as for example from 300 s to 350 s.

To give a better insight into the performance and to come up with a fair comparison of calibrations, two intervals are investigated in detail. Therein, the controller has not been deactivated during the measurement. Altogether the proposed method reliably yields applicable controller parameters. Additionally, all stages, including the initial design, do not show any unstable or unexpected behavior.

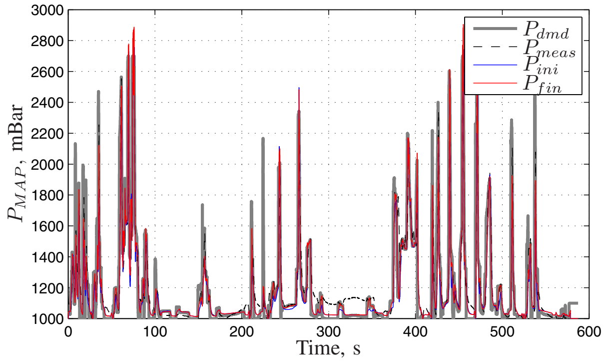

Fig. 8. Comparison of measured reference performance and simulation of the controller calibration stages: Overall

Detail A, which shows the range from 50 s to 120 s is depicted in Figure 9. Already the initial design achieves equal performance as the conventional in-vehicle calibration. The final calibration further improves the performance.

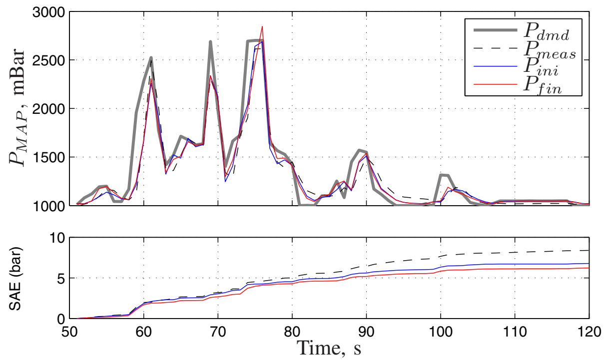

Fig. 9. Comparison of measured reference performance and simulation of the controller calibration stages: Detail A

Figure 10 (Detail B) illustrates the range from 350 s to 550 s. It shows similar results as compared to Detail A.

A quantitative comparison of the conventional calibration and the two calibration stages is given in Table I. For this purpose, the final cumulated absolute errors E of the different calibrations according to the time intervals shown in Figures 8-10 are compared. In addition, the relative improvement compared to the measured reference performance is given.

	Detail A		Detail B	
E_{meas}	8.3915	100%	18.342	100%
E_{ini}	6.7912	80.9%	15.148	82.6%
E_{fin}	6.2204	74.1%	13.528	73.8%

TABLE I

QUANTITATIVE COMPARISON OF THE CUMULATED ABSOLUTE ERROR E (IN BAR) AND ITS RELATIVE VALUE COMPARED TO THE MEASURED REFERENCE PERFORMANCE E_{meas}

This application example shows, that the introduced calibration method achieves better controller performance than the conventional manual calibration.

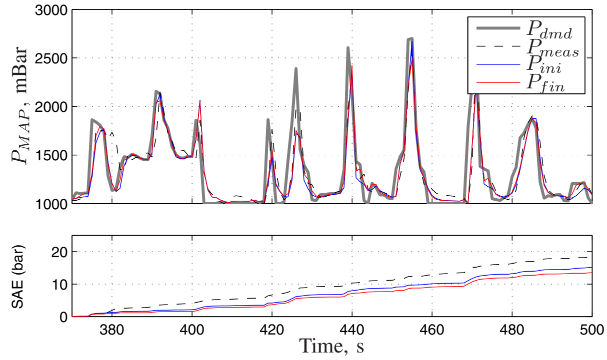

Fig. 10. Comparison of measured reference performance and simulation of the controller calibration stages: Detail B

VI. Conclusion

A new calibration method for ECU PID controllers is introduced in this paper. The proposed method is model-based and offers an efficient and robust basic calibration. However, fine-tuning will still be carried out by calibration engineers during test rides, because subjective driving experience is not reproducible by computer aided calibration methods.

Acknowledgment

This work was supported by the Christian Doppler Research Association and AVL List GmbH.

References

[1] R. Murray-Smith and T. A. Johansen, "Multiple Model Approaches to Modelling and Control," *Taylor & Francis, London*, 1997.

[2] S. Jakubek and C. Hametner, "Identification of Neurofuzzy Models Using GTLS Parameter Estimation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 5, pp. 1121–1133, Oct. 2009.

[3] A. Leva, C. Cox, and A. Ruano, *Hands-on PID autotuning: a guide to better utilisation*. IFAC Professional Brief, 2002.

[4] K. J. Åström and T. Hägglund, *Advanced PID control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2006.

[5] M. Nebel, M.-S. Vogels, T. Combe, H. Pfluegl, T. Winsel, and C. Hametner, "Global dynamic models for xil-based calibration," in *SAE 2010 world congress and exhibition. Session: Engine control and calibration (Parts 1 of 2)*, Detroit, MI, USA, April 2010.

[6] C. Hametner and S. Jakubek, "Local model network identification for online engine modelling," *Information Sciences*, vol. 220, pp. 210–225, 2013.

[7] C. Hametner and M. Nebel, "Operating regime based dynamic engine modelling," *Control Engineering Practice*, vol. 20, no. 4, pp. 397 – 407, 2012.

[8] C. Hametner, C. Mayr, M. Kozek, and S. Jakubek, "PID controller design for nonlinear systems represented by discrete-time local model networks," *International Journal of Control*, vol. 86, no. 9, pp. 1453–1466, 2013.

[9] C. Mayr, C. Hametner, M. Kozek, and S. Jakubek, "Nonlinear Stable PID Controller Design using Local Model Networks," in *Proceedings of the 20th Mediterranean Conference on Control and Automation*, Jul. 2012, pp. 842–847.

[10] C. Hametner and S. Jakubek, "New Concepts for the Identification of Dynamic Takagi-Sugeno Fuzzy Models," in *IEEE Conference on Cybernetics and Intelligent Systems*, Jun. 2006, pp. 185–190.

[11] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Cinchester: John Wiley & Sons Ltd., 2009.

[12] I. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, 2003.