European Journal of Control 64 (2022) 100606

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

Optimal plug-and-control of unknown nonlinear systems

Daniel Adelberger[a,*], Laura Giarré[b,c], Toshiyuki Ohtsuka[d], Luigi del Re[a]

[a] Johannes Kepler University (JKU), Linz, Austria
[b] DIEF, Universitá di Modena e Reggio Emilia, Modena, Italy
[c] IIR, NTNU, Ålesund, Norway
[d] Department of Systems Science, Graduate School of Informatics, Kyoto University, Kyoto, Japan

ARTICLE INFO

Article history:
Received 31 May 2021
Revised 1 October 2021
Accepted 26 December 2021
Available online 7 January 2022

Recommended by Prof. T Parisini

Keywords:
Optimal control
Nonlinear identification
Nonlinear control
Adaptive control

ABSTRACT

Modeling and control design are typically subsequent but independent activities. Optimal control is traditionally developed on the basis of explicit models. While this usually yields good results for linear systems, the same is not as true for nonlinear ones, for which explicit solutions can be found only for few cases. In practice, in most cases receding horizon controls based on linear approximations are used. In this paper, we propose a procedure which delivers in one step both a model and an optimal receding horizon control algorithm, without requiring a linearization. Our procedure relies essentially on a system identification by a suitable class of functions which offers universal approximation properties that can be directly incorporated in the control algorithm. Using directional forgetting we show that an adaptive extension can be realized. Measurements and simulations based on a standard automotive control problem are presented to confirm the validity of our proposal.

© 2022 The Author(s). Published by Elsevier Ltd on behalf of European Control Association.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

Optimal operation is a common requirement for control systems, and optimal control is an important part of it. Optimal control is a topic whose different facets have been studied over a very long period (see for example [7]). An optimal control usually results from two sequential steps, first setting up a model and then using optimal control theory to obtain the control algorithm for the given requirements.

While for linear systems there are well established methods which work with the utmost majority of cases, the same is not as true for nonlinear systems, for which many results exist, but no general procedure. One of the reasons is that the analytical formulation of optimal control of nonlinear systems can lead to very complex and non-convex expressions [33], which are very seldom solvable explicitly, and often very difficult to solve even numerically, at least in real time. Indeed, many proposals have been formulated to explicitly take into account nonlinearity, providing approximated solutions for some specific classes, just as an example see [27], but there is no universal solution. Also embedded compensation of the nonlinearity has been proposed, see e.g. [14,23]. In their majority, these works rely on state space, even continuous system description, which may be a major handicap for unknown nonlinear systems, for which nonlinear identification provides discrete-time input/output models for which a state space continuous time form may not even exist. Using input/output measurements for data-enabled predictive control is well known, see e.g. [9]. However, these methods are mainly concerned with linear systems [3,4]. Also the excitation conditions are not simple to define, a critical point when discussing adaption and forgetting strategies.

Many nonlinear system identification methods have been proposed, see e.g. [32], until more recently [19,20]. However, these methods usually require many data and there is no established procedure to design parsimonious experiments, even though some proposals exist (see e.g. [25]). This makes their use difficult in our context as measurements, for instance in the case of our example, are expensive and slow.

The idea behind this paper is to look for a practical method able to cope with a generally unknown nonlinear system described only by sampled measurements as they are, i.e. without any linearization or transformation into a state space (SS) and/or continuous time form. In order to simplify the design, we also look for a procedure in which the result of the modeling phase can be inserted immediately into an suitable optimizing algorithm, in the usual form of a receding horizon form.

* Corresponding author.
E-mail addresses: daniel.adelberger@jku.at (D. Adelberger), laura.giarre@unimore.it (L. Giarré), ohtsuka@i.kyoto-u.ac.jp (T. Ohtsuka), luigi.delre@jku.at (L. del Re).

https://doi.org/10.1016/j.ejcon.2021.12.001
0947-3580/© 2022 The Author(s). Published by Elsevier Ltd on behalf of European Control Association. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

Our solution is based on identifying polynomial nonlinear autoregressive models with exogenous input (PNARX models) following our own fast strategy as described in [31] and using the result in a modified version of the continuation/generalized minimal residual (C/GMRES) method [21], a truly nonlinear model predictive control algorithm which has been extended to work with input/output (I/O) time discrete models [5]. PNARX are of course not the only possible choice, but they offer some very interesting advantages – universal approximators, linear-in-parameters and trivial differentiability.

Its key advantage are its speed and general applicability: the experimental examples – on a very complex system – shown below have taken a few hours from scratch to online-operation. It also yields a better performance compared to the industrial production solution, which is the result of a huge design and tuning work.

As always, there is also a price to be paid. This is mainly the potentially large number of parameters of the model, a common problem to all universal approximators. This means that the excitation needed to identify a model can be difficult to obtain. Accordingly, our method consists of an offline and an online step. In the offline step, we us an iterative approach, based on a combination of design of experiment (DOE) phases, which tune the information content of the data, as well as of pruning phases, which remove unnecessary model parts. For the online operation, the excitation cannot be optimized by DOE, instead we propose using a directional forgetting to allow an adaptive operation.

While some partial results have been published earlier by the authors, the aim of this paper is to present the final framework as well as its practical use and performance. The rest of the paper is organized as follows. In Section 2 our approach is introduced, in Section 3 the used nonlinear system is described and modeled, in Section 4 the offline and online identification method is reported together with the controller and finally in Section 5 the discrete optimal controller is described and the results are presented in Section 6. Some conclusion are drawn in Section 7.

2. Optimal plug and control of nonlinear systems

2.1. The goal

Our interest consists in developing a method by which an optimal control for a (partly) unknown nonlinear system can be developed in one step, by experiments to be performed in an initial phase. Additionally, we are interested in providing some degree of adaptation to be able to track changes as for instance due to wear during normal operation.

2.2. The problem

As is generally known, a generic nonlinear system can be described in its continuous time state space form by

$$\dot{x} = f(x, u, t) \tag{1a}$$

$$y = h(x, u, t), \tag{1b}$$

where $x \in \mathbb{R}^n$ denotes the state, $u \in \mathbb{R}^m$ the input, t the time, and $y \in \mathbb{R}^l$ the output. With a scalar cost function J

$$J(t_1) = \Phi(x(t_1), t_1) + \int_{t_0}^{t_1} \varphi(x, u, t)\mathrm{d}t, \tag{2}$$

where t_0 and t_1 are the initial and final time, $\varphi(\cdot)$ the running cost, and $\Phi(t_1)$ the terminal cost, the standard optimal control problem formulation is

$$u(t) = \arg\min_u J \tag{3}$$

s.t. $\dot{x} = f(x, u, t)$.

It is well known that, by application of Bellman's optimality principle [17] under some conditions this optimal control problem can be reformulated by the Hamilton–Jacobi–Bellman (HJB) equation as the search of an optimal function $V^*(x, t)$ which has to fulfill a partial differential equation. If a nonlinear input affine system

$$\dot{x} = f(x) + g(x)u \tag{4}$$

and a quadratic cost function in u is considered, e.g.

$$J(t_1) = \int_{t_0}^{t_1} q(x) + u^{\mathrm{T}} R u \, \mathrm{d}t \tag{5}$$

this partial differential equation becomes

$$-\frac{\partial V^*}{\partial t} = q(x) + \frac{\partial V^*}{\partial x} f(x) - \frac{1}{4} g(x)^\top \frac{\partial V^*}{\partial x}^\top \frac{\partial V^*}{\partial x} g(x) \tag{6}$$

and an explicit expression for the control can be derived

$$u^*(x) = -\frac{1}{2} R^{-1} g(x)^\top \frac{\partial V^*}{\partial x}^\top.$$

Unfortunately, solving the partial differential equation for V^* is usually not possible, except for very special functions $f(x)$ and $g(x)$ and special $q(x)$, for example for linear systems with a quadratic cost function. Otherwise, possibly strong and mostly ad-hoc approximations are needed, see e.g. [26,27].

2.3. Our proposal

Against this background, we suggest that both parts of the control design process, the modeling and the control design, should be coordinated, in particular, the modeling should be conceived as the "best" approximation of the input/output behavior of the nonlinear system by a class for which a solution to the optimal control problem is directly available. Indeed, modeling is to some extent always an approximation process, as no model can ever represent a system completely. First principle modeling is the method of choice for many applications, in particular for simulations, but such models require sufficient knowledge about the system to be modeled, tend to be complex and to include elements which will be anyway removed during the approximation steps needed for control design.

System identification is one method of modeling, essentially the mapping of the actual I/O behavior of a system on a class of basic functions. If the system is not known sufficiently well to choose a specific class, a class of universal approximators can be used which allow a good approximation of the I/O behavior (see [8]). Unfortunately, not all universal approximators yield models which can be directly used for optimal control design (see e.g. [6,10,35]).

With that in mind, we propose the combination of polynomial NARX models [16,22] and C/GMRES control [21], as the latter does not need any linearization, so that the parameters of the PNARX model can be used directly. As the model depends only on data, in principle it can be updated.

All this leads to an adaptive optimal control according to the scheme that is sketched in Fig. 1.

It should be noticed, however, that in order to realize this structure two challenges must be met. In particular, universal approximators tend to yield a large number of parameters which make the identification much more difficult. Against this background, we suggest a two-step procedure: in an offline step, we identify the full model, while in the online situation we only update it according to the available excitation. For the offline phase, the authors have proposed an offline iterative identification framework in [31] which uses DOE steps to obtain a simpler model with good prediction properties. For the online phase, a DOE is not possible, but using directional forgetting allows to update at least parts of

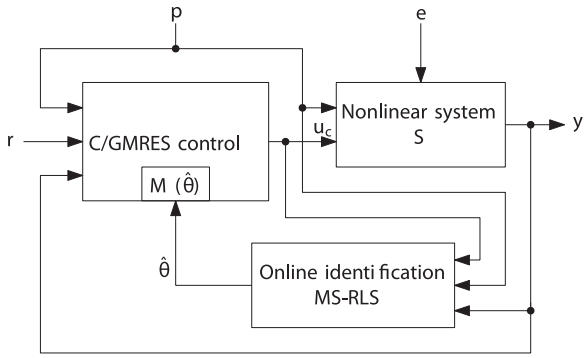

Fig. 1. Scheme of the adaptive optimal control.

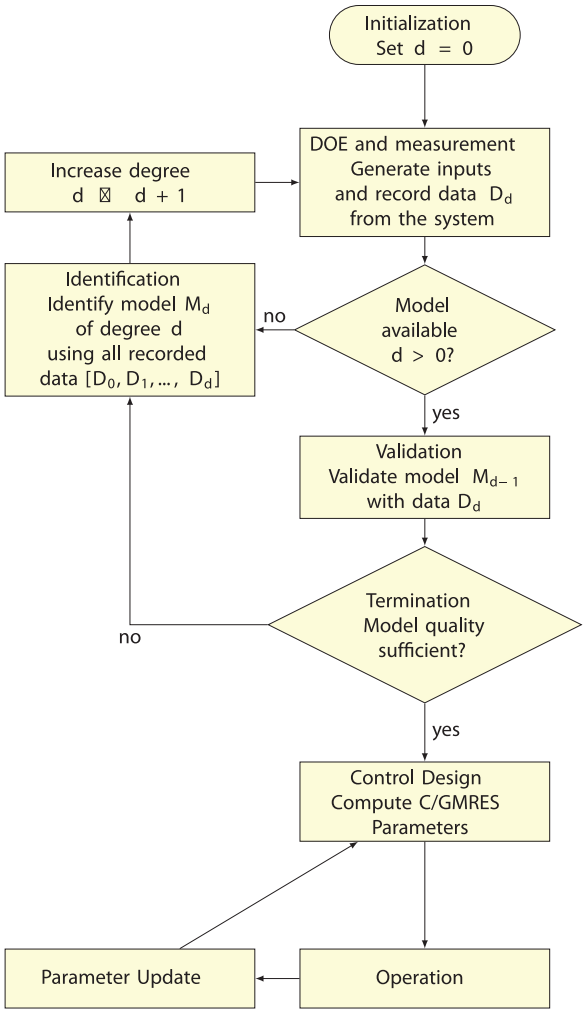

Fig. 2. Meta-algorithm describing the proposed procedure.

the model whenever a sufficient excitation is available. This is adequate for most practical applications, because usually a few properties change over time.

The other much simpler challenge is adapting the C/GMRES method, which has been developed for continous time state space models to the framework of identification – i.e. discrete time input/output models – which was already presented by the authors in [5].

The final design procedure steps of the *plug and control adaptive scheme* whose components will be described in details in the following, can be represented by a meta-algorithm as shown in Fig. 2:

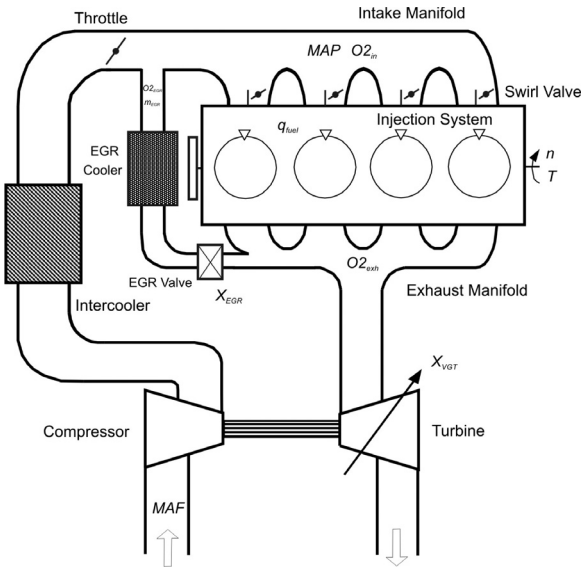

Fig. 3. Schematic of the airpath.

In order to assess the performance of our method, we shall use a very well known problem of automotive control, the air path control of a Diesel engine.

3. Test application: air path control

In an internal combustion engine ideal combustion happens if the oxygen-fuel ratio is exactly in balance. In practice an optimal combustion is not possible and therefore exhaust gases like nitrogen oxides (NO_x), carbon monoxide (CO) as well as particle matters (PM) are produced in addition to the products of an ideal combustion, carbon dioxide (CO_2) and water (H_2O). CO occurs only under few conditions, and PM can be prevented from reaching the atmosphere by filters, so the key interest is in NO_x. A well established method to reduce NO_x consists in recirculating parts of the exhaust gases, mixing them with the fresh air which arrives to the combustion chamber. This recycling is done via the exhaust gas recirculation (EGR) valve. In addition to that modern engines can also have a turbocharger with variable geometry turbine (VGT), which is used to increase the total amount of fresh air led to the combustion chamber. Both control inputs EGR valve position and VGT guide vane position can be used to effectively reduce harmful emissions if properly controlled. The physical background for the possible emission reduction can be found e.g. in [15]. A schematic of an airpath equipped with a VGT turbocharger and external high pressure EGR is shown in Fig. 3.

Engine control is mainly a feed-forward control, in which set points are decided at a higher level and then control loops enforce them. In the case of air path control, typically manifold air pressure (*MAP*) and fresh air to the manifold (mass air flow – *MAF*) are the dynamic set points. The control problem becomes then

$$\min_{MAP(t),\,MAF(t)} \int q_1 (MAP(t) - MAP_{\text{ref}}(t))^2 + \\ + q_2 (MAF(t) - MAF_{\text{ref}}(t))^2 \text{dt}.$$

Fig. 4 shows the experimental setup, consisting of a N47 2 liters engine by BMW, a highly dynamic brake by AVL and the corresponding measurement technique (see https://desreg.jku.at for details). Tests are done with the production engine control unit (ECU) by Bosch, where some functions can be bypassed so to modify only the desired output to the actuators. A vehicle simulation software, in our case CarMaker/InMotion [2], was used to calculate the load

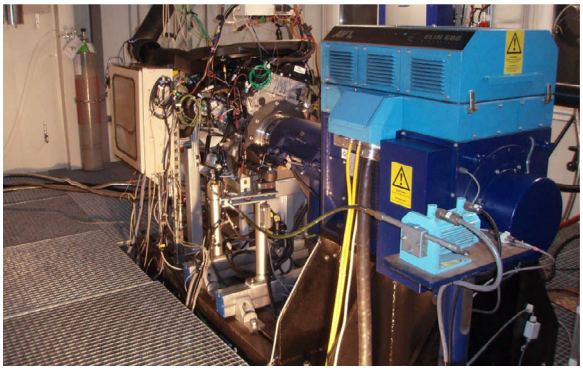

Fig. 4. Experimental setup of the engine test bench with a BMW N47 Diesel engine.

Table 1
Inputs and outputs of the air path model.

Description	Symbol	Unit	Value range	
			min.	max.
Inputs				
EGR valve position	X_{EGR}	%	0	100
VGT guide vane position	X_{VGT}	%	60	95
Measured Disturbances				
Engine speed	n_{e}	min^{-1}	800	3000
Injected fuel per cycle	m_{f}	mg	0	45
Outputs				
IM air mass flow per cycle	*MAF*	mg	0	1800
IM air pressure	*MAP*	mbar	800	3500

conditions so that the engine would experience the very same load as on the road. The key advantage is the reproducibility, because road measurement happen always in different conditions, while the conditions in the test cell, in particular humidity and temperature of the intake air and temperature of the surrounding air are kept constant. Additionally, many more measurements are possible in a test cell than in practice.

Usually, only the position of the VGT and the EGR can be manipulated, as the injected fuel amount and the actual speed of the internal combustion engine are given by the dynamics of the drive chain and the driver. The resulting multiple input multiple output (MIMO) model (from VGT and EGR to *MAP* and *MAF*) is highly nonlinear, so a dynamic feed-forward control of the model is quite hard to realize due to the nearly impossible inversion of the systems behavior. In the industrial practice, this is approximately achieved by very complex map based local control loops.

Notice that engines are operated in limited operating regions, see Table 1 for our values.

4. Adaptive nonlinear identification

4.1. Offline identification

Recall that the function

$f_d\colon \mathbb{R}^n \to \mathbb{R}^{l(n,d)}$,

where $l(n,d) = \frac{1}{n!}\prod_{i=0}^{n-1}(d+i+1)$, is denoted polynomial space basis function of degree $d \in \mathbb{N}$. With $\boldsymbol{x} \in \mathbb{R}^n$, $n \in \mathbb{N}$, being

$\boldsymbol{x} = [x_1\ x_2\ x_3\ \cdots\ x_n]^\top$,

it can be defined recursively such that

$$f_0(\boldsymbol{x}) = 1, \quad f_1(\boldsymbol{x}) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} f_{1,0}(\boldsymbol{x}) \\ f_{1,1}(\boldsymbol{x}) \\ f_{1,2}(\boldsymbol{x}) \\ \vdots \\ f_{1,n}(x) \end{bmatrix}, \tag{7}$$

and in general for $i \geq 2$

$$f_i(\boldsymbol{x}) = \begin{bmatrix} \begin{bmatrix} f_{i-1,0}(\boldsymbol{x}) \\ \vdots \\ f_{i-1,n}(\boldsymbol{x}) \end{bmatrix} \cdot 1 \\ \begin{bmatrix} f_{i-1,1}(\boldsymbol{x}) \\ \vdots \\ f_{i-1,n}(\boldsymbol{x}) \end{bmatrix} \cdot x_1 \\ \begin{bmatrix} f_{i-1,2}(\boldsymbol{x}) \\ \vdots \\ f_{i-1,n}(\boldsymbol{x}) \end{bmatrix} \cdot x_2 \\ \vdots \\ \begin{bmatrix} f_{i-1,n}(\boldsymbol{x}) \end{bmatrix} \cdot x_n \end{bmatrix} = \begin{bmatrix} f_{i,0}(\boldsymbol{x}) \\ f_{i,1}(\boldsymbol{x}) \\ f_{i,2}(\boldsymbol{x}) \\ \vdots \\ f_{i,n}(\boldsymbol{x}) \end{bmatrix}. \tag{8}$$

a PNARX model results. As it is linear in parameters, it can be rewritten in the same form as usual for linear models

$$y_k = f_d(\boldsymbol{x}_k)^\top \boldsymbol{\theta} + e_k = \boldsymbol{\varphi}_k^\top \boldsymbol{\theta} + e_k, \tag{9}$$

with

$$\boldsymbol{x}_k^\top = \begin{bmatrix} y_{k-n} & \dots & y_{k-1} & u_{k-m} & \dots & u_k \end{bmatrix}. \tag{10}$$

albeit with the difference that the unknown parameters are multiplied by a function of the measured values and not the values themselves. Again, writing (9) for all k from 1 to N yields the equation system

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\theta} + \boldsymbol{e}, \tag{11}$$

The price to be paid for using PNARXs – as with universal approximators in general – is the possible explosion of the number of parameters. Of course, not all parameters are as important for the quality of the model, and there are methods to keep their number increase under control. It must also be noticed that, in practice, the approximations quality will be limited not only by the model complexity, but also by the properties of $\boldsymbol{e}$. See [31] for more details.

4.2. Data based modeling of the airpath

In order to identify a data based model, a DOE procedure is applied in the first place to receive sufficient excitation required for an initial parameter identification.

For the DOE the approach described in [13] was used and signals of length 1000s have been recorded. The DOE involves scaling of the designed signals, for which the input signal ranges in Table 1 are used. There are several DOE optimality criteria available, in this work we employ D-optimal input design, which is often used because the determinant criterion is invariant to linear scaling and re-parametrization of models. D-optimal input design can be seen as minimizing the volume of the confidence ellipsoid of the parameter covariance matrix of the identified model. The iterative algorithm developed in [36] and also used in [13] was utilized to generate an approximately D-optimal input sequence.

In Fig. 5 the results obtained after applying the DOE signal of degree 4 are shown. Note that there is a slight deviation between desired (setpoint; red) signals applied to the ECU and the actual measured signals (blue) obtained by the sensors/ECU.

4.3. Offline model identification

For the identification, the model orders have been selected to $m_{\text{u}} = 3$ and $m_{\text{y}} = 2$ for the first output (*MAF*) and $m_{\text{u}} = 5$ and $m_{\text{y}} = 1$ for the second output (*MAP*). These parameters were tuned empirically based on experience. A polynomial degree of $d_{\text{p}} = 2$

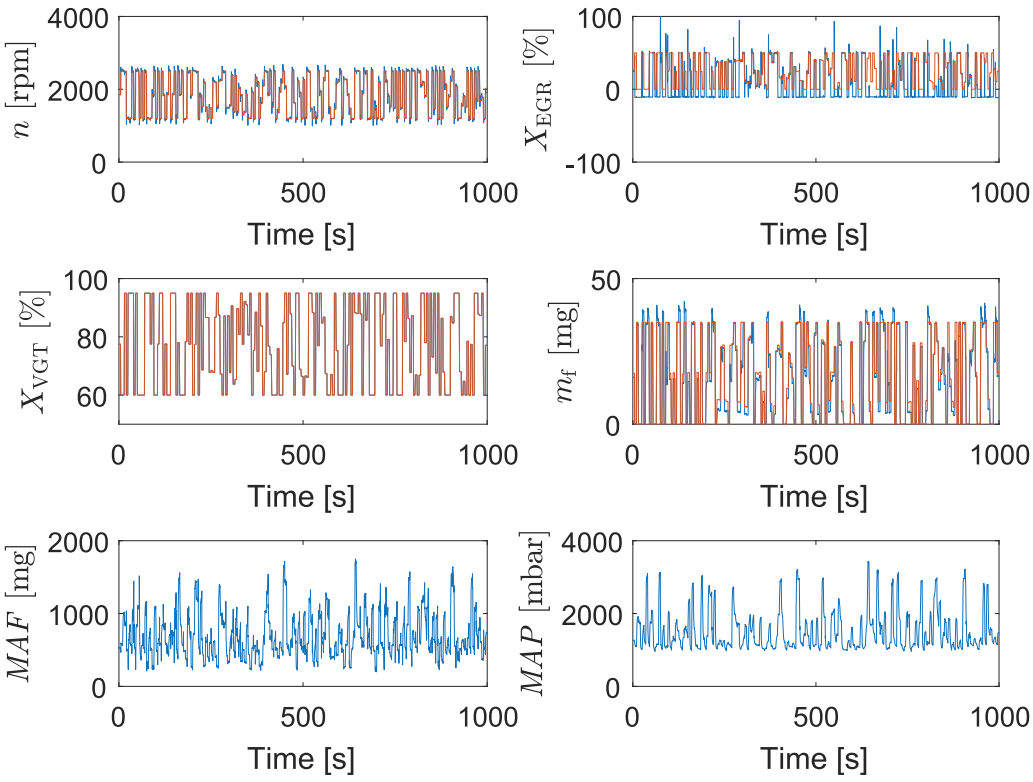

Fig. 5. Input and output signals of the engine air path for DOE inputs for a model of degree 4.

was used for both model outputs. The possible number of polynomial regressors has been pruned to 30 regressors for *MAF* and 29 regressors for *MAP* by a LASSO-based regressor selection algorithm (see [34]) using the identification data in order to avoid overfitting. This leads to a total of 59 model parameters to be identified. The identification with the previously shown DOE data of degree 4 yielded superior results compared to DOE data of lower degree as well as WLTP data (where the Worldwide harmonized Light vehicles Test Procedure was completed instead of a specific DOE), obtained by cross validation between the respective data sets. In detail, all identified models have been validated using the other datasets and the 1-step prediction error as well as the simulation error (see [18]) utilizing the FIT value, defined in [13, Eq. (4.88)]. Exemplary results for the setting of $n = 2$ identification steps and the WLTP data (assumed to be a realistic validation data set) are shown in Fig. 6.

It can be observed, that the outputs of the plant with an optimized parameter set θ^* (yellow lines in Fig. 6) approximate the behavior of the airpath system quite well.

4.4. Online identification

In [29] two ways of suitable online identification for systems are presented, namely a recursive least squares (RLS) algorithm with exponential forgetting (RLS-EF) as well as a RLS algorithm with directional forgetting (RLS-DF). Both methods shall be shortly revised here. As extensively shown in [29] the algorithms are robust to the noise model structure (see the numerical example sessions in [29]). Due to lack of space, we do not repeat such examples here.

By application of the RLS algorithm the parameter-vector $\hat{\theta}$ can be estimated online to be utilized for the approximation of the unknown system $\mathcal{S}$ by the PNARX model $\mathcal{M}(\hat{\theta})$. This enables tracking of a reference r by $\mathcal{S}$ without further knowledge of the actual nonlinear system, utilizing the discrete-time I/O C/GMRES method presented in Section 5.2.

The identification data is defined using N measurements from time step $k_s = k$ to $k_e = k + N - 1$ (where the (n) is dropped in the following since it is regarded a fixed number)

$$\Phi_N^i = \begin{bmatrix} \varphi^i(n,k_s)^\top \\ \vdots \\ \varphi^i(n,k_e)^\top \end{bmatrix} = \begin{bmatrix} \varphi_{k_s}^{i\ \top} \\ \vdots \\ \varphi_{k_e}^{i\ \top} \end{bmatrix} \quad Y_N^i = \begin{bmatrix} y_{k_s}^i \\ \vdots \\ y_{k_e}^i \end{bmatrix} \tag{12}$$

and the (parameter dependent) prediction vector by

$$\hat{Y}_N^i(\theta) \quad = \begin{bmatrix} \hat{y}_{k_s|k_s-n}^i \\ \vdots \\ \hat{y}_{k_e|k_e-n}^i \end{bmatrix} = \Phi_N^i \cdot \Theta^i(n,\theta) = \Phi_N^i \cdot \Theta^i(\theta). \tag{13}$$

for which – to be precise – n additional measurements are needed. Note that Φ_N^i is a $(N \times m_i)$ matrix and Y_N^i is a $(N \times 1)$ vector. Denote $m_t = \sum_{i=1}^{m_y} m_i$, so $\theta \in \mathbb{R}^{m_t}$.

The quadratic error of the n-step prediction model is used as a cost function to get the optimal parameters θ_N, as

$$J_N(\theta) = \sum_{i=1}^{m_y} (Y_N^i - \hat{Y}_N^i(\theta))^\top W_N (Y_N^i - \hat{Y}_N^i(\theta)), \tag{14}$$

$$\theta_N = \arg\min_\theta J_N(\theta). \tag{15}$$

In order to perform exponential forgetting of old data the weighting matrix W_N is chosen as

$$W_N = \mathrm{diag}(w_N), \qquad w_N = \begin{bmatrix} \lambda^{N-1} \\ \vdots \\ \lambda^{N-N} \end{bmatrix} = \begin{bmatrix} \lambda^{N-1} \\ \vdots \\ \lambda^0 \end{bmatrix}, \tag{16}$$

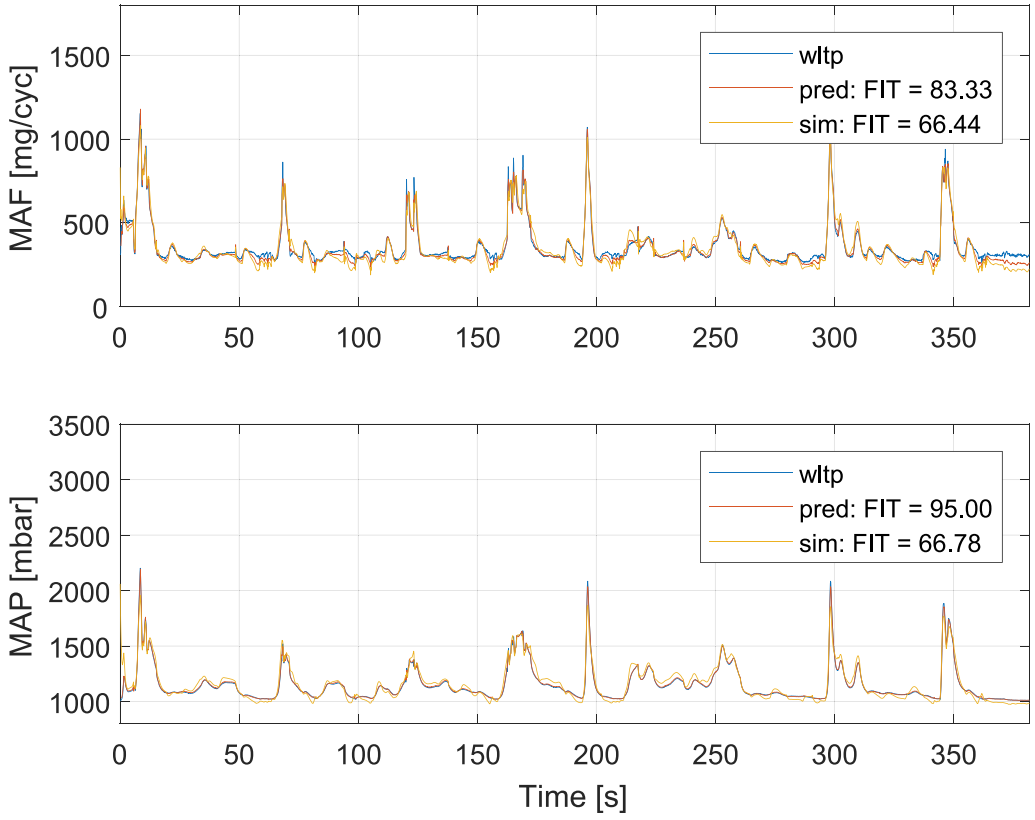

Fig. 6. Validation result (1-step prediction and simulation) for WLTP data and $n = 2$ step identification.

where $\lambda \in [0, 1]$ is the forgetting factor. If $\lambda = 1$ is chosen, the case without forgetting is obtained.

In [30] the recursive algorithm for (approximately) solving this problem is derived [30, Algorithm 1]. In this algorithm the parameter update is done via a Newton-step

$$\theta_N = \theta_{N-1} - \mathcal{H}(\theta_{N-1}, g_N, H_N)^{-1}\mathcal{G}(\theta_{N-1}, g_N, H_N) \tag{17}$$

with

$$\mathcal{G}(\theta_{N-1}, g_N, H_N) = \left(\frac{\partial J_N(\theta)}{\partial \theta}\right)^{\top}\Bigg|_{\theta=\theta_{N-1}} \tag{18}$$

$$= -2\sum_{i=1}^{m_y}\Theta^{i'}(\theta_{N-1})\left[g_N^i - H_N^i\Theta^i(\theta_{N-1})\right] \tag{19}$$

$$\mathcal{H}(\theta_{N-1}, g_N, H_N) = \left(\frac{\partial}{\partial\theta}\mathcal{G}(\theta, g_N, H_N)\right)\Bigg|_{\theta=\theta_{N-1}} \tag{20}$$

$$\Theta^{i'}(\theta) = \left(\frac{\partial}{\partial\theta}\Theta^i(\theta)\right)^{\top} \tag{21}$$

and the tuples

$$g_N = (g_N^1, \ldots, g_N^{m_y}), \qquad H_N = (H_N^1, \ldots, H_N^{m_y}). \tag{22}$$

The recursive exponential forgetting update strategy for g_N and H_N is

$$g_N^i = \lambda g_{N-1}^i + \varphi_N^i y_N^i \tag{23}$$

$$H_N^i = \lambda H_{N-1}^i + \varphi_N^i \varphi_N^{i\top} \tag{24}$$

for each $i \in \{1, \ldots, m_y\}$.

According to [29], it is possible to incorporate also directional forgetting in this n-step prediction identification scheme. The key idea of directional forgetting is to forget only past data in the direction of the new incoming information.

Let's first consider the 1-step RLS algorithm,

In the 1-step case the function $\Theta(\theta) = \theta$ is the identity function and therefore $\Theta'(\theta) = I$ is the identity matrix. Then

$$\mathcal{G}(\theta_{N-1}, g_N, H_N) = -2[g_N - H_N\theta_{N-1}]$$
$$\mathcal{H}(\theta_{N-1}, g_N, H_N) = 2H_N.$$

4.4.1. Recursive parameter update with g_N

The recursive parameter update becomes

$$\begin{aligned}\theta_N &= \theta_{N-1} + \frac{1}{2}(H_N)^{-1}2[g_N - H_N\theta_{N-1}]\\ &= \theta_{N-1} + (H_N)^{-1}g_N - (H_N)^{-1}H_N\theta_{N-1}\\ &= (H_N)^{-1}g_N\end{aligned}$$

This is the standard RLS algorithm without matrix inversion lemma. Note that this algorithm converges after one iteration, since

$$\begin{aligned}\mathcal{G}(\theta_N, g_N, H_N) &= -2[g_N - H_N\theta_N]\\ &= -2\left[g_N - H_N(H_N)^{-1}g_N\right] = 0.\end{aligned}$$

4.4.2. Recursive parameter update without g_N

The standard form of the RLS without matrix inversion lemma can be derived by eliminating g_N. Recall (23) and (24) for the scalar case $m_y = 1$

$$g_N = \lambda g_{N-1} + \varphi_N y_N \tag{25}$$

$$H_N = \lambda H_{N-1} + \varphi_N\varphi_N^{\top}. \tag{26}$$

$$\theta_N = H_N^{-1} g_N = H_N^{-1}(\lambda g_{N-1} + \varphi_N y_N) \tag{27}$$

$$H_N\theta_N = \lambda g_{N-1} + \varphi_N y_N. \tag{28}$$

By using $\theta_{N-1} = H_{N-1}^{-1} g_{N-1}$ and therefore $g_{N-1} = H_{N-1}\theta_{N-1}$ we get

$$H_N\theta_N = \lambda H_{N-1}\theta_{N-1} + \varphi_N y_N \tag{29}$$

By plugging in (26) for λH_{N-1} we get

$$H_N\theta_N = (H_N - \varphi_N\varphi_N^\top)\theta_{N-1} + \varphi_N y_N \tag{30}$$

and therefore with

$$\theta_N = H_N^{-1}(H_N - \varphi_N\varphi_N^\top)\theta_{N-1} + H_N^{-1}\varphi_N y_N \tag{31}$$

$$= \theta_{N-1} + H_N^{-1}\varphi_N(y_N - \varphi_N^\top\theta_{N-1}) \tag{32}$$

the standard RLS form.

By making use of the results in [30], the corresponding recursive update of g_N is obtained as:

$$\theta_N = \theta_{N-1} + H_N^{-1}\varphi_N(y_N - \varphi_N^\top\theta_{N-1}) \tag{33}$$

$$= H_N^{-1}\underbrace{(H_N\theta_{N-1} + \varphi_N y_N - \varphi_N\varphi_N^\top\theta_{N-1})}_{g_N} \tag{34}$$

With $\theta_{N-1} = H_{N-1}^{-1} g_{N-1}$:

$$g_N = H_N H_{N-1}^{-1} g_{N-1} + \varphi_N y_N - \varphi_N\varphi_N^\top H_{N-1}^{-1} g_{N-1}. \tag{35}$$

By plugging in the recursive update of H_N, we get

$$\begin{aligned} g_N &= (H_{N-1} - (1-\lambda)\Delta H_N + \varphi_N\varphi_N^\top)H_{N-1}^{-1} g_{N-1} + \\ &\quad \varphi_N y_N - \varphi_N\varphi_N^\top H_{N-1}^{-1} g_{N-1} \\ &= g_{N-1} - (1-\lambda)\Delta g_N + \varphi_N y_N \end{aligned} \tag{36}$$

$$\Delta g_N = \Delta H_N H_{N-1}^{-1} g_{N-1} = \Delta H_N \theta_{N-1}. \tag{37}$$

4.5. Multi-step directional forgetting RLS

Combining the parameter update from (17), (19) and (20), with the update for g_N and H_N, we obtain the general algorithm for dealing with both directional forgetting and exponential forgetting (Algorithm 1) with the tuples defined in (22), y_N^i the ith element

Algorithm 1 Multi-Step Recursive LS.

```
procedure MS-RLS(θ_{N-1}, g_{N-1}, H_{N-1}, y_N, φ_N, λ, ε)
    for i ∈ {1, ..., m_y} do
        g_N^i = g_{N-1}^i − (1 − λ)Δg_N^i + φ_N^i y_N^i
        H_N^i = H_{N-1}^i − (1 − λ)ΔH_N^i + φ_N^i φ_N^{iT}
    end for
    g_N ← (g_N^1, ..., g_N^{m_y})
    H_N ← (H_N^1, ..., H_N^{m_y})
    θ_N ← θ_{N-1}
    while ‖G(θ_N, g_N, H_N)‖ > ε do
        θ_N ← θ_N − H(θ_N, g_N, H_N)^{-1} G(θ_N, g_N, H_N)
    end while
    return (θ_N, g_N, H_N)
end procedure
```

of the vector y_N and the tuple $\varphi_N = (\varphi_N^1, \ldots, \varphi_N^{m_y})$.

For *directional forgetting*, the update is

$$\Delta H_N^i = \begin{cases} \frac{H_{N-1}^i \varphi^i(N)\varphi_N^{iT} H_{N-1}^i}{\varphi_N^{iT} H_{N-1}^i \varphi_N^i} & \text{if } \|\varphi_N^i\| \geq \epsilon \\ 0 & \text{if } \|\varphi_N^i\| < \epsilon \end{cases} \tag{38}$$

$$\Delta g_N^i = \Delta H_N^i (H_{N-1}^i)^{-1} g_{N-1}^i. \tag{39}$$

Note that the *exponential forgetting* case can also be represented in this form by using the update

$$\Delta H_N^i = H_{N-1}^i, \qquad \Delta g_N^i = g_{N-1}^i. \tag{40}$$

which can be interpreted as uniform forgetting in "all directions" of H_{N-1}^i.

5. Input/output discrete time C/GMRES

5.1. Basic algorithm

Nonlinear model predictive control has been attracting the attention of researchers already for decades. Correspondingly, there are very many available proposals we cannot review here for sake of space, see e.g. [12,15] and references therein.

C/GMRES [21] is a somewhat different method because it essentially concentrates on solving the Euler-Lagrange equations at some time instants over a limited time horizon in a moving horizon fashion.

To recall the key idea, let us consider again the general system of (1a) under some equality constraints

$$C(x(t), u(t), p(t)) = 0. \tag{41}$$

where $p(t) \in \mathbb{R}^p$ denotes a measured disturbance. Inequality constraints can also be transformed into the equality constraints by introducing quadratic slack variables and by adding linear penalties on the slack variables in the cost function [21]. Another practical method is to use barrier functions with fixed barrier parameters. It is well known that, with (2) a Hamiltonian can be defined as

$$\begin{aligned} &H(x(t), u(t), \lambda(t), \mu(t), p(t)) \\ &\quad = \varphi(x(t), u(t), p(t)) + \lambda^T(t) f(x(t), u(t), p(t)) \\ &\quad + \mu^T(t) C(x(t), u(t), p(t)), \end{aligned} \tag{42}$$

where $\lambda(t)$ indicates the co-states and $\mu(t)$ the Lagrange multiplier of the equality constraint. Over the chosen horizon, the necessary conditions for an optimum at the sample times can be rewritten in vectorial form as

$$\begin{aligned} x_{k+1} &= x_k + T_s f(x(t), u(t), p(t)) \\ x(k=0) &= x_0 \\ \lambda_k^{*T} &= \lambda_{k+1}^{*T} + \frac{\partial H}{\partial x_k^*} T_s \\ \lambda_K^{*T} &= \frac{\partial \varphi}{\partial x_k}|_{x_k = x_K} + \mu_K^{*T}\frac{\partial C}{\partial x_k^*}|_{x_k^* = x_k^*} \\ F(U(t), x(t), t) &= \begin{bmatrix} \frac{\partial H(x_0^*, u_0^*, \lambda_1^*, \mu_0^*, p_0^*)}{\partial u_0} \\ C(x_0^*, u_0^*, p_0^*) \\ \cdots \\ \frac{\partial H(x_{K-1}^*, u_{K-1}^*, \lambda_K^*, \mu_{K-1}^*, p_{K-1}^*)}{\partial u_{K-1}} \\ C(x_{K-1}^*, u_{K-1}^*, p_{K-1}^*) \end{bmatrix} \\ &= 0, \end{aligned} \tag{43}$$

where $U(t)$ indicates the sequence of inputs to the system over the prediction horizon. As finding the zeroes of the nonlinear function $F(U(t), x(t), t)$ is no easy task, the continuation method [24] can be used. It essentially consists in replacing (and relaxing) the search of the zeros, which could be done, e.g., by Newton's method, by enforcing $F(U(t), x(t), t)$ to approach zero as time increases. Of course, all problems related to the possible non-convexity remain, e.g., the starting point must lie in space of attraction of the solution. In order to use the continuation method,

in a first step, the product of Jacobians with vectors is rewritten as:

$$\begin{aligned}\dot{F}(U,x,t) &= \frac{\partial F}{\partial U}\dot{U}+\frac{\partial F}{\partial x}\dot{x}+\frac{\partial F}{\partial t}\\ &\approx \frac{F(U+\dot{U}h,x+\dot{x}h,t+h)-F(U,x,t)}{h}\\ &:= D_hF(U,x,t;\dot{U},\dot{x},1),\end{aligned} \tag{44}$$

with a positive real number h. Using this leads to

$$\begin{aligned}&D_hF(U,x+\dot{x}h,t+h;\dot{U},0,0)\\ &= A_sF(U,x,t)-D_hF(U,x,t;0,\dot{x},1).\end{aligned} \tag{45}$$

This is a linear equation concerning $\dot{U}$, which can be solved quite fast by using the GMRES algorithm. The algorithm is based on the Krylov-subspace method, which is designed to solve a large linear equation. The advantage of this algorithm is that the residual is reduced monotonically during iterations and a solution of a sufficient accuracy is obtained with a much less number of iterations than the dimension of the unknown quantity $\dot{U}(t)$.

5.2. Algorithm for discrete-time I/O model

In this paper, the system is assumed to have the form

$$x_{k+1}=P(x_k,x_{k-n_a+1},u_{k-n_k},u_{k+1-n_k-n_b},p_{k-n_{pk}},u_{k+1-n_{pk}-n_b}), \tag{46}$$

where n_a indicates the maximum number of time shifts of the outputs, n_k the delay time till the input effects the output and n_b the maximum number of the inputs effecting the output. n_{pk} indicates the time shifts of the time dependent parameters of the system, till they occur at the output of the system and n_p is the maximum number of the time dependent parameters effecting the output of the system. In the following, (46) is shortened to $x_{k+1}=P(x_k,x_{k-n_a},u_{k-n_k},u_{k-n_k-n_b+1})$ to improve readability, but the time dependent parameters of the system are still included in the calculations.

The aim is again to minimize the performance index J over K prediction steps to get the optimal control input u_k^* for the system to be controlled. To solve this problem the principles of static optimization can be used to get the necessary conditions. First, the Lagrange-function related to the equations stated above

$$\begin{aligned}L' &= J+\mu_k^TC(x_k,u_k,p_k)\\ &\quad+\lambda_{k+1}^T(P(x_k,x_{k-n_a},u_{k-n_k},u_{k-n_k-n_b+1})-x_{k+1})\end{aligned} \tag{47}$$

is introduced.

By using the necessary conditions, the equations

$$\begin{aligned}\frac{\partial L'}{\partial u_k} &= 0=\frac{\partial L(x_k,u_k,p_k)}{u_k}+\mu_k^T\frac{\partial C(x_k,u_k,p_k)}{\partial u_k}\\ &\quad+\sum_{i=k+n_k}^{\max(k+n_k+n_b,K)}\lambda_{i+1}\frac{\partial P(x_i,x_{i-n_a},u_{i-n_k},u_{i-n_k-n_b+1})}{\partial u_k}\end{aligned} \tag{48a}$$

$$\begin{aligned}\frac{\partial L'}{\partial x_k} &= 0=\frac{\partial L(x_k,u_k,p_k)}{x_k}+\mu_k^T\frac{\partial C(x_k,u_k,p_k)}{\partial x_k}\\ &\quad+\sum_{i=k-1}^{\max(k+n_a,K)}\lambda_{i+1}\frac{\partial(P(x_i,x_{i-n_a},u_{i-n_k},u_{i-n_k-n_b+1})-x_{i+1})}{\partial x_k}\end{aligned} \tag{48b}$$

$$\frac{\partial L'}{\partial x_k}|_{k=K}=0=\mu_K^T\frac{\partial C}{\partial x_k}|_{k=K}+\frac{\partial\varphi}{\partial x_k}|_{k=K} \tag{48c}$$

hold in an optimal point, where $\mu\in\mathbb{R}^{p\times K}$ indicates the vector of the Lagrange-multipliers for the equality constraints and $\lambda\in\mathbb{R}^{n\times K}$ the vector of the Lagrange-multipliers of the systems equations, holds. These equations are similar to those of Pontryagin's minimum principle, but now, especially (48b), for each k is linked to some others. That means in general not only two equations are linked via the time shifted variables. So it is necessary, to have a closer look at them. Let's start at the terminal equation for λ_K:

$$\lambda_{K+1}=\mu_K^T\frac{\partial C}{\partial x_k}|_{k=K}+\frac{\partial\varphi}{\partial x_k}|_{k=K} \tag{49}$$

In this equation, all variables can be predicted. The backwards calculation should be no problem using (48b), because it is legal to cut off the summation if the index exceeds K, because it is assumed that K is chosen in a way making the system able to reach the desired output within. Otherwise, the costs of the penalty state are changing λ_{K+1}. Due to this, in step $K-1$ the summation only lasts over two indices and again all variables can be calculated. To make the equations fit to the C/GMRES algorithm, they have to be reformulated,

$$\begin{aligned}x_{k+1} &= P(x_k,x_{k-n_a},u_{k-n_k},u_{k-n_k-n_b+1})\\ \lambda_{k+k'} &= \frac{\partial L'}{\partial x_{k+k'}}\\ \lambda_{k+K} &= \frac{\partial\varphi}{\partial x_{k'}^*}|_{k'=k+K}+\mu_{k'}^{*T}\frac{\partial C}{\partial x_{k'}^*}|_{k'=k+K}\\ F(U_k,x_k,k) &= \begin{bmatrix}\frac{\partial L'(x_k^*,u_k^*,\lambda_{k+1}^*,\mu_k^*,p_k^*)}{\partial u_k}\\ C(x_k^*,u_k^*,p_k^*)\\ \vdots\\ \frac{\partial L'(x_{k+K-1}^*,u_{k+K-1}^*,\lambda_{k+K}^*,\mu_{k+K-1}^*,p_{k+K-1}^*)}{\partial u_{k+K-1}^*}\\ C(x_{k+K-1}^*,u_{k+K-1}^*,p_{k+K-1}^*)\end{bmatrix}\\ &= 0.\end{aligned} \tag{50}$$

Based on (50), the equation $F(U_k,x_k,k)=0$ can still be solved using the continuation method, as proposed in [5] for a discrete-time state equation. To this end, it is necessary to introduce the difference between two time steps using the sampling time T_s

$$\begin{aligned}\Delta x_k &= \frac{x_{k+1}-x_k}{T_s}\\ \Delta U_k &= \frac{U_{k+1}-U_k}{T_s}\end{aligned} \tag{51}$$

by means of a forward differentiation. This results in

$$\begin{aligned}\Delta F(U_k,x_k,k) &= A_sF(U_k,x_k,k)=\\ &= \frac{\partial F(U_k,x_k,k)}{\partial x_k}\Delta x_k+\frac{\partial F(U_k,x_k,k)}{\partial U_k}\Delta U_k\\ &\quad+\frac{F(U_k,x_k,k+1)-F(U_k,x_k,k)}{T_s}.\end{aligned} \tag{52}$$

Compared to (44) and (45) this is equal to $D_hF(U,x,t;W,w,\omega)$, if $W=\frac{\partial U_k}{\partial k}$, $w=\frac{\partial x_k}{\partial k}$ and $\omega=1$ is assumed. Under these assumptions, the C/GMRES algorithm can be executed similarly to the continuous-time case. The necessary modifications are:

- $D_hF(U,x+h\dot{x},t+h;\dot{U},0,0)$ becomes $\frac{F(U_k+\Delta U_kT_s,x_{k+1},(k+1)T_s)-F(U_k,x_{k+1},(k+1)T_s)}{T_s}$
- $A_sF(U,x,t)-D_hF(U,x,t;0,\dot{x},1)$ becomes $A_sF(U_k,x_k,k)-\frac{F(U_k,x_{k+1},(k+1)T_s)-F(U_k,x_k,kT_s)}{T_s}$

It is possible to determine these quantities within each sampling step, if the prediction for U in the former timestep was accurate enough. Since the nonlinear model predictive control algorithm realizes feedback control, it inherently suppresses the influence of noises. The overall method works satisfactorily even under measurement noises in the real experiment, as shown in Section 6.

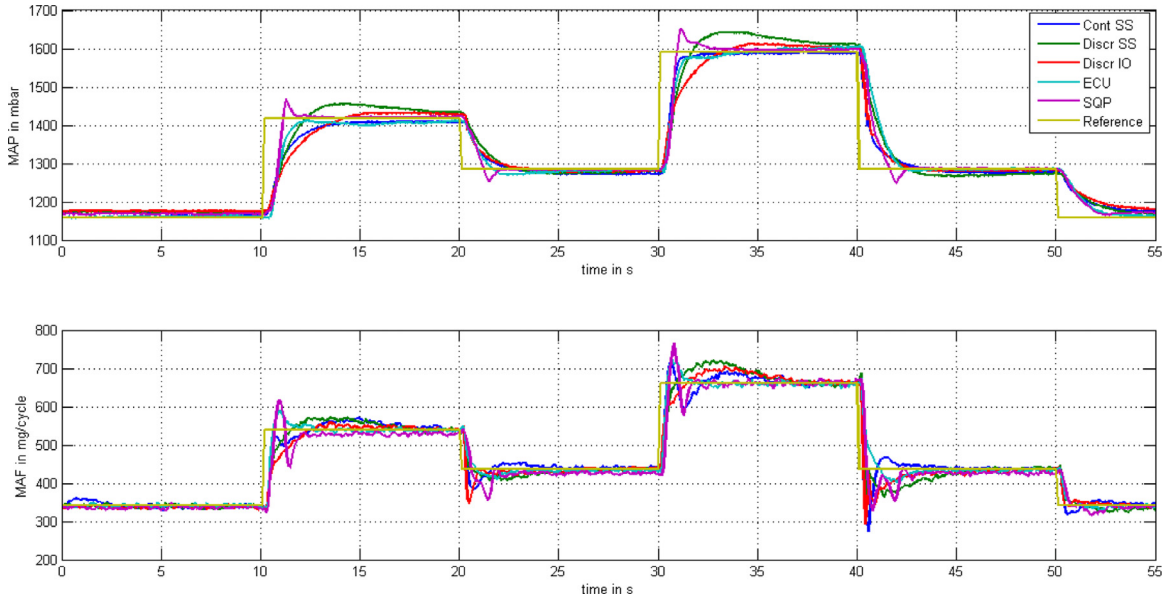

Fig. 7. Comparison of the application of the algorithm with artificial trajectory.

6. Results

6.1. Offline method

A sensible approach to test the performance of the method consists in comparing it to the standard production implementation, which is the result of an enormous experimental optimization work using dedicated tools like AVL CAMEO [1]. Engine control is mainly a feed-forward control, and in particular air path control is traditionally separated in two steps: first suitable reference values are defined, which achieve the overall requirements, like consumption and emissions, and then a lower level feedback mainly on the actuators is used to enforce them. In dynamic transients, it is critically to track the references as precisely as possible. Fig. 7 shows the performance of the production version (ECU) and of the suggested non adaptive algorithm (Discr IO) in time domain. Fig. 8 shows the distance from the estimated Pareto front. It may surprise that the distance from the Pareto front is roughly the same. Actually, the opposite would be true, behind the performance of the ECU there are may working years of optimization work while the Discr IO results derive from a few hours tuning. Indeed, if the problem is left unchanged, the optimum will be the same, so no improvement in performance can be expected, the optimum is just the same, whatever method is used to approximate it.

Both figures, however, show also other three cases, meant to assess the impact of the method, of the discretization, and of the use of a input/output form instead of a state space one. Our PNARX model can be linearized at every step and used for a standard sequential quadratic programming (SQP). The time plot in Fig. 7 shows a slower behavior, but the Pareto plot is clearly showing a substantial drop in performance. The PNARX model can be transformed into a higher dimensional state space form [11] and a very fast Euler approximation can be used for control. The results are given as Cont SS in both plots. The performance (in terms of distance from the Pareto front), is the same as the production form.

Finally, this very state space form can be discretized in a standard way. This leads to the performance of Discr SS.

Following conclusions seem important:

- Using a nonlinear control method (C/GMRES) proves better than a linearized one.
- Discrete time approximation leads to a loss of performance, but the form state space or input/output does not make a substantial difference.
- The work amount form problem to solution goes from hours to person years.

Of course, it is an example, but consistent with our general experience, and gives a good reason to suggest this method.

6.2. Nominal control

A nonlinear model predictive control (NMPC) for tracking reference profiles of *MAF* and *MAP* (which is a standard approach and also done e.g. in the former works [5,11,28]) is designed using the identified 2-step model. The previously described C/GMRES is used to efficiently solve the receding-horizon optimal control problem. The NMPC controls only the inputs $u_c = [X_{EGR}, X_{VGT}]^\top$, while the measured disturbances $p = [n_e, m_f]^\top$ are given by the requirement of the driver.

We use a similar scenario as in [28], given by (almost) constant signals $n_e = 2000\,\text{rpm}$ and $m_f = 20\,\text{mg/cycle}$ shown in Fig. 10 and reference trajectories for *MAF* and *MAP* as filtered step sequences shown in Fig. 9. The controller is applied to the identified model in simulation and in the nominal setting described in this subsection the controller has perfect model parameter knowledge. The input to the plant is denoted $u = [u_c, p]^\top$ (including measured disturbances), the output is denoted $y = [MAF, MAP]^\top$ and the reference signals denoted $r = [MAF_{ref}, MAP_{ref}]^\top$. The cost function of the controller at time step k' is defined as

$$J_{MPC} = \varphi(\Delta y(k' + n_{PH})) + T_s \sum_{k=k'}^{k'+n_{PH}} L(\Delta y(k), \Delta u(k)) \tag{53}$$

with $\Delta y(k) = y(k) - r(k)$, $\Delta u(k) = u_c(k) - u_c(k-1)$, the terminal cost function

$$\varphi(\Delta y) = \Delta y^\top S \Delta y, \tag{54}$$

and the running cost function

$$L(\Delta y, \Delta u) = \Delta y^\top Q \Delta y + \Delta u^\top R \Delta u, \tag{55}$$

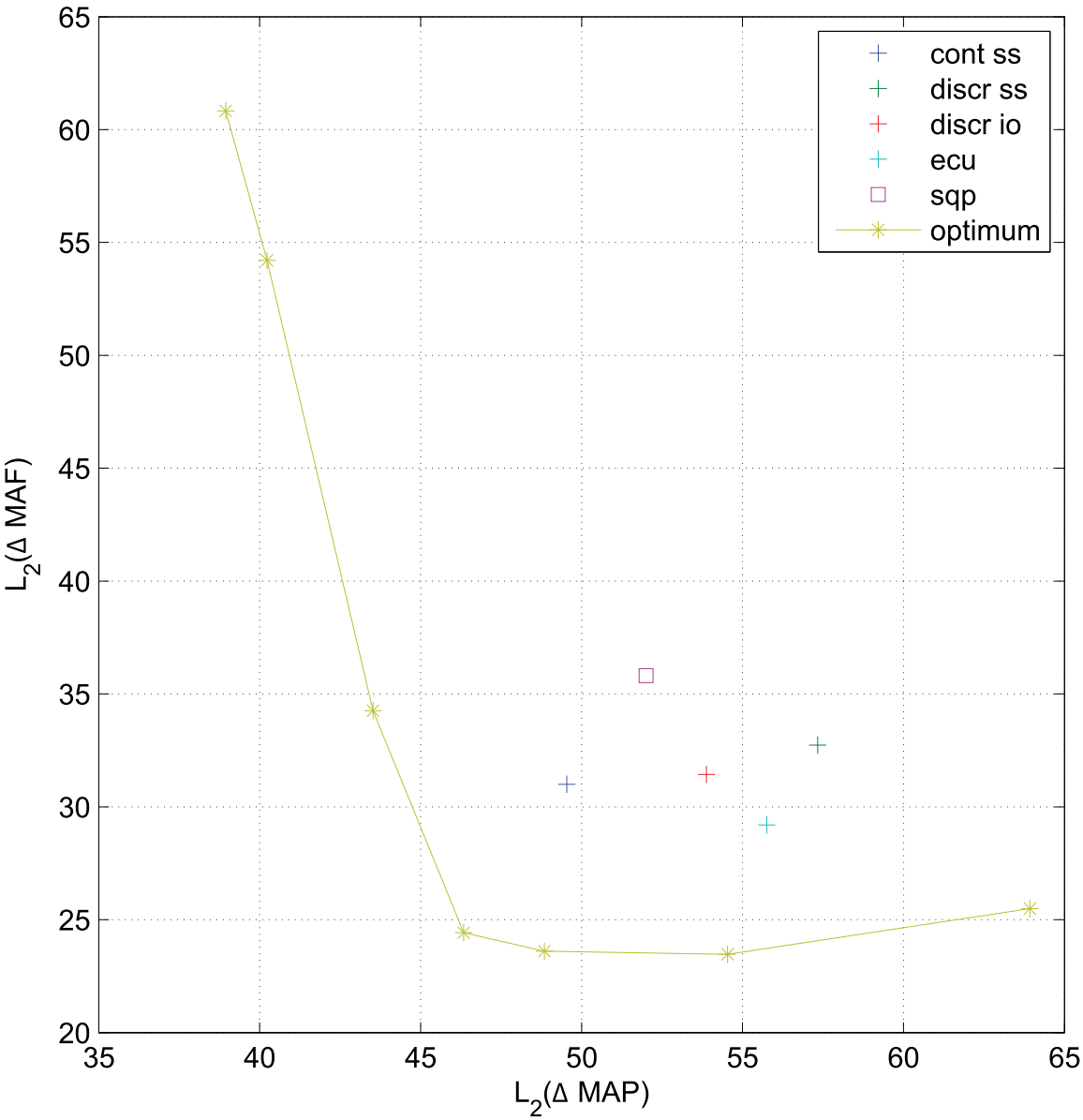

Fig. 8. Performance in terms of distance from the Pareto optimal solution.

where the weighting matrices of this quadratic cost function have been set to

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}, S = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}. \tag{56}$$

For evaluating the control performance over the whole scenario of $T = 40s$ the evaluation cost function is defined as

$$J = T_s \sum_{k=0}^{T/T_s} L(\Delta y(k), \Delta u(k)). \tag{57}$$

The following parameters have been used for the C/GMRES algorithm: prediction horizon $n_{\mathrm{PH}} = 10$, max. iterations of GMRES $k_{\max} = 10$, step size $h = 0.002$, stabilization parameter $\zeta = 1/h$, relative tolerance $r_{\mathrm{tol}} = 1 \times 10^{-6}$, no preconditioning, no look-ahead.

This nominal control, where the model parameters in the prediction model $\hat{\theta}$ are the same as the plant model parameters θ^*, leads to the results depicted in Figs. 9 and 10. As expected, the controlled outputs fit really well to the provided reference.

6.3. Perturbation of the model used in controller

Now the sensitivity of the control performance with respect to a deviation in model parameters (used only in the prediction model of the NMPC) is examined. With this trick, the contrary – namely the behavior of the controller in case of an arbitrary deviation in system parameters can be emulated. This is not realized directly, as the plant with optimal parameter set θ^* realizes a realistic air path behavior (as visible in Fig. 6) and the parameter set in the NMPC should always converge to such a realistic parameter set (even if they are disturbed to represent an unrealistic behavior) – which is shown in the following. The plant remains the same (parameters θ^*), but the prediction model parameters $\hat{\theta}$ in the controller are disturbed

$$\hat{\theta} = \theta^*(1 + \varepsilon \Delta\theta) \tag{58}$$

where $\Delta\theta$ is a random parameter vector from a uniform distribution in $[-1, 1]$ and ε is the gain of the disturbance. Variations up to 5% of the nominal value have been made, i.e. $\varepsilon \in [-0.05, 0.05]$. The resulting loss in control performance is shown by evaluation

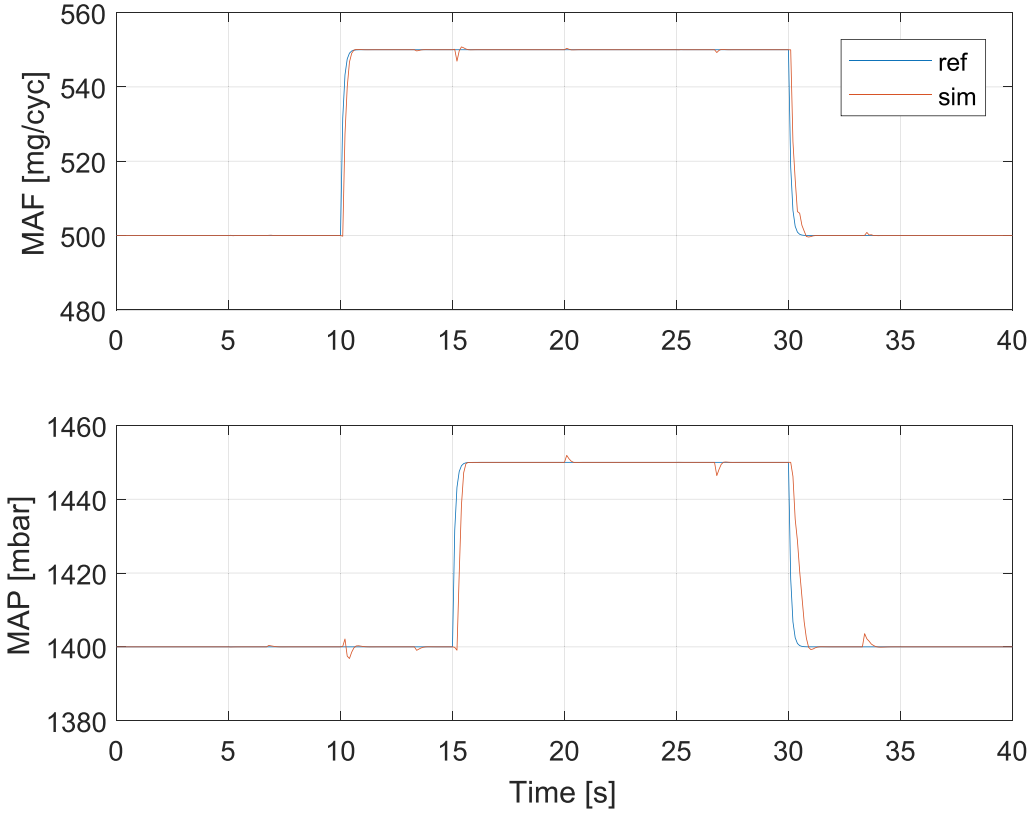

Fig. 9. Reference signals (blue) and system outputs (red) of the nominal controller applied to the identified model in simulation.

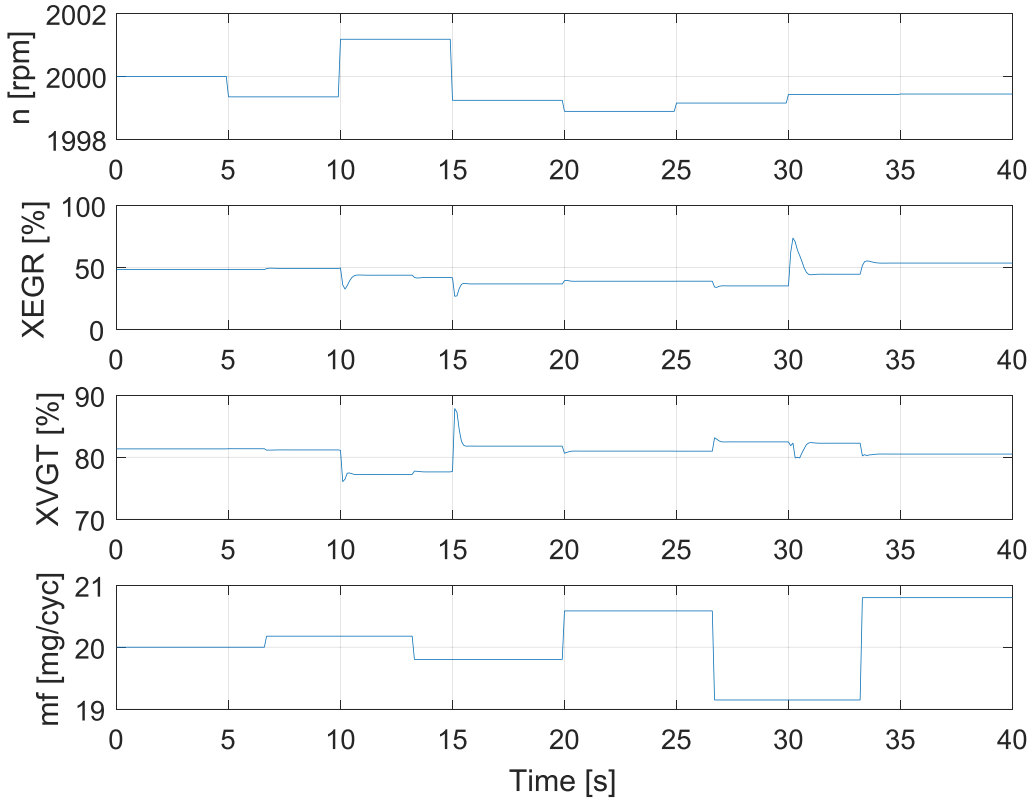

Fig. 10. Measured disturbances (n_e, m_f) feedforward-applied to the plant and control inputs (X_{EGR}, X_{VGT}) resulting from the nominal NMPC control.

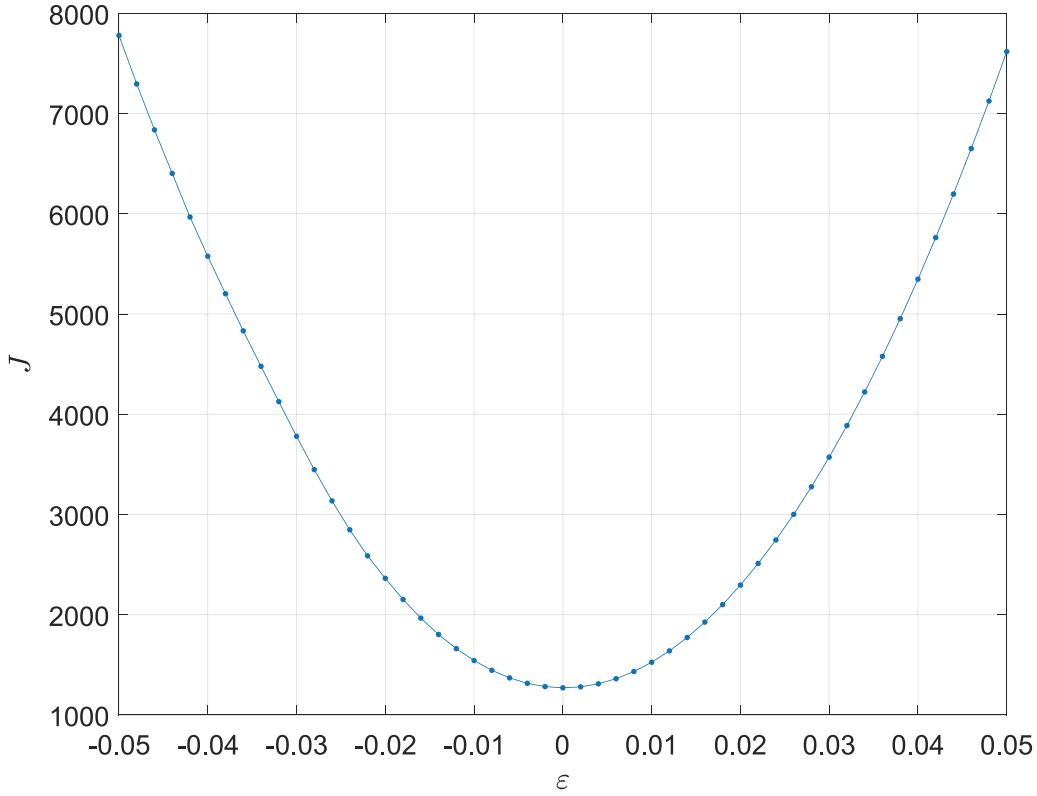

Fig. 11. Sensitivity analysis of the control performance with respect to a perturbation of the prediction model parameters in the controller, represented by a gain ε.

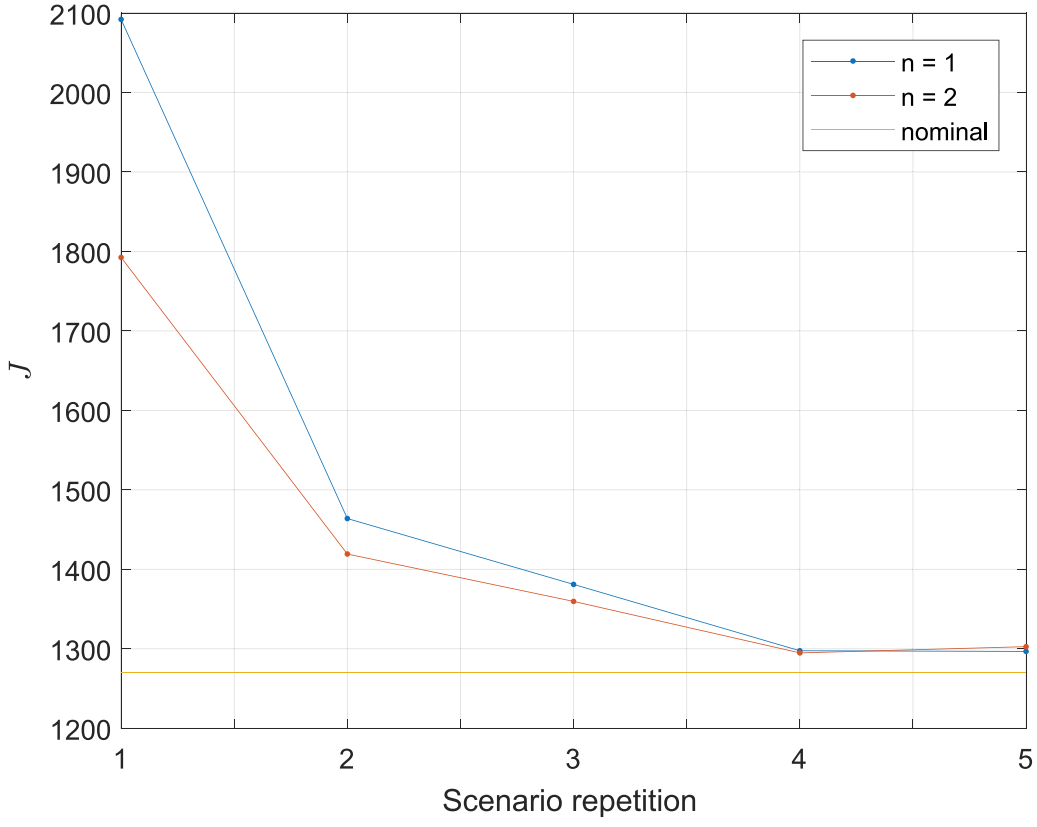

Fig. 12. Comparison of performance of online identification with nominal control (best case).

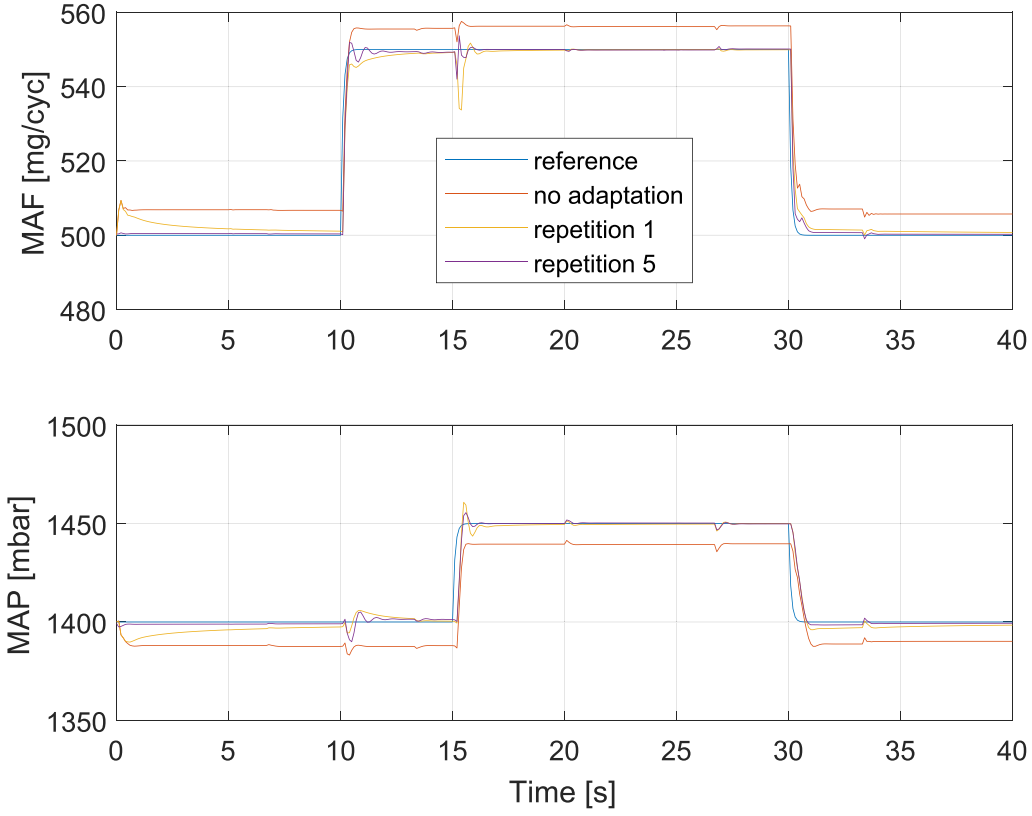

Fig. 13. Tracking signals comparison without and with adaptation ($n = 2$).

the cost function J with respect to ε, which is shown in Fig. 11. It can be seen that there is a high sensitivity and at the maximum performance loss (where $\varepsilon = -0.05$) the cost function value is $J = 7779$, about 6 times higher as in the nominal case.

6.4. Adaptive expansion

The previously defined NMPC is now combined with the multi-step recursive LS (MS-RLS) algorithm with directional forgetting (Algorithm 1) in order to get an adaptive, self-tuning NMPC controller that is able to react to initially wrong model parameters or slowly changing system parameters over time. Directional forgetting is used in order to cope with the low information of the identification signals typically appearing in a tracking closed-loop application, as discussed in [29]. A comparison between $n = 1$ and $n = 2$ for the parameter estimation in the MS-RLS algorithm is made. The control scheme is depicted in Fig. 1.

The scenario (reference signals r and disturbance p from Figs. 9 and 10) has been repeated 5 times while applying the self-tuning NMPC continuously for this longer scenario. In this way the cost function J can be evaluated 5 times. The result of this simulation is shown in Fig. 12, where MS-RLS settings of $n = 1$ and $n = 2$ are compared to the nominal case cost (flat line).

A parameter deviation with $\varepsilon = -0.05$ was used as initial parameters in the C/GMRES prediction model and MS-RLS (for which no adaptation leads to a cost function value of $J = 7779$). The following settings were used in the MS-RLS algorithm: forgetting factor $\lambda = 0.999$, directional forgetting threshold $\epsilon = 1 \times 10^{-3}$ [29, Eq. (28)], tolerance (ε in Algorithm 1 of [29]) of 1×10^{-6}, maximum iterations 100.

Fig. 13 shows exemplary results of the tracking outputs: the reference is compared with the case no adaptation ($\hat{\theta}$ constant with $\varepsilon = -0.05$), and the first and fifth repetition of the case with adaptation (MS-RLS with $n = 2$ and settings described above). It is shown that the tracking performs better with adaptation and after several repetitions the performance is close to the nominal case.

In the presented closed-loop case, however, the parameters estimated by MS-RLS do not yet converge to the true parameters of the plant, because the signals n_e and m_f are in a very narrow range and the algorithm with directional forgetting has too little information to estimate the parameters correctly.

7. Conclusions

Optimality is a common requirement for most industrial systems, and optimal control a key element for it. While the theory is well established, its practical application is mostly limited to linear system as there is no sufficiently general approach for nonlinear optimal control. Approximations are always needed, either the model needs to be simplified or the control task, most both of them.

Control design is usually based on models, for nonlinear systems as well, and the key suggestion of this paper is to tailor the model to a suitable control system design method. What this paper shows is that the combination of PNARX models and C/GMRES does provide such an environment for many practical cases, and that this environment can be extended in direction of adaptive control. Of course, approximations are still needed, but they happen in a systematic way which requires little or no intervention by the user.

As every control method, physical boundaries are not moved. The modeling phase will come to a limit as soon as stochastic aspects become important – something very relevant in the examples mentioned here. The optimum will be the same independently on the way is reached – the control performance of the optimal tracking of the *MAP*/*MAF* in the example is not significantly different from the production solution. The difference, however, is the price: behind the industrial example we have huge amounts of

tuning work on an extremely complex control structure, the same results are achieved in few hours with a simple structure in our case. The difference becomes even more important for the large number of applications for which the tuning effort of an engine control unit would not be affordable – engines are produced in millions – so that the optimal performance would not be reached. The key price to be paid is the curse of dimensionality, which in practice means that a completely adaptive version will hardly be feasible, as no input will be exciting enough. We also need access to the plant, and the ability to perform some experiments – most medical applications, for instance, would exclude it. Still, the approach should bring substantial benefits for most industrial applications.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work has been supported by the LCM K2 Center within the framework of the Austrian COMET-K2 program.

References

[1] AVL CAMEO(https://www.avl.com/-/avl-cameo-4-a), Accessed: 2020-10-06.
[2] Product brochure avl inmotion 4.0
[3] J. Berberich, A. Koch, C.W. Scherer, F. Allgöwer, Robust data-driven state-feedback design, in: Proceedings of the American Control Conference (ACC), IEEE, 2020a, pp. 1532–1538.
[4] J. Berberich, J. Köhler, M.A. Müller, F. Allgöwer, Data-driven model predictive control with stability and robustness guarantees, IEEE Trans. Autom. Control 66 (4) (2020b) 1702–1717.
[5] J. Blumenschein, T. Schwarzgruber, R. Schmied, T.E. Passenbrunner, H. Waschl, L. del Re, Approximate optimal control of discrete I/O systems with C/GMRES, in: Proceedings of the European Control Conference (ECC) 2015, Linz, Austria, EUCA, 2015, pp. 104–110.
[6] D. Bruder, X. Fu, R. Vasudevan, Advantages of bilinear Koopman realizations for the modeling and control of systems with unknown dynamics, IEEE Robot. Autom. Lett. 6 (3) (2021) 4369–4376, doi:10.1109/LRA.2021.3068117.
[7] A.E. Bryson, Y.-C. Ho, Applied optimal control: optimization, estimation, and control, 1975.
[8] N.E. Cotter, The stone-weierstrass theorem and its application to neural networks, IEEE Trans. Neural Netw. 1 (4) (1990) 290–295.
[9] J. Coulson, J. Lygeros, F. Dörfler, Data-enabled predictive control: in the shallows of the deepc, in: Proceedings of the 18th European Control Conference (ECC), IEEE, 2019, pp. 307–312.
[10] M. Ekman, Suboptimal control for the bilinear quadratic regulator problem: application to the activated sludge process, IEEE Trans. Control Syst. Technol. 13 (1) (2005) 162–168, doi:10.1109/TCST.2004.838556.
[11] D. Gagliardi, T. Othsuka, L. del Re, Direct C/GMRES control of the air path of a Diesel engine, in: Proceedings of the IFAC World Congress 2014, 2014.
[12] L. Grüne, J. Pannek, Nonlinear Model Predictive Control: Theory and Algorithms, Springer International Publishing, Cham, 2017, doi:10.1007/978-3-319-46024-6_1.
[13] M. Hirsch, Identification of a Virtual Sensor Model for Diesel Engine Emissions by Means of Optimal Input Design, Advances in Mechatronics, Trauner Verlag, 2012.
[14] M. Huang, H. Nakada, S. Polavarapu, R. Choroszucha, K. Butts, I. Kolmanovsky, Towards combining nonlinear and predictive control of diesel engines, in: Proceedings of the American Control Conference, 2013, pp. 2846–2853, doi:10.1109/ACC.2013.6580266.
[15] T.A. Johansen, Introduction to nonlinear model predictive control and moving horizon estimation, Sel. Top. Constr. Nonlinear Control 1 (2011) 1–53.
[16] I.J. Leontaritis, S.A. Billings, Input-output parametric models for non-linear systems Part I: deterministic non-linear systems, Int. J. Control 41 (2) (1985) 303–328.
[17] F.L. Lewis, D. Vrabie, V.L. Syrmos, Optimal Control, John Wiley & Sons, 2012.
[18] L. Ljung, System Identification - Theory for the User, Prentice Hall PTR, 1999.
[19] B. Lusch, J.N. Kutz, S.L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, Nat. Commun. 9 (1) (2018) 1–10.
[20] D. Masti, A. Bemporad, Learning nonlinear state–space models using autoencoders, Automatica 129 (2021) 109666.
[21] T. Ohtsuka, A continuation/GMRES method for fast computation of nonlinear receding horizon control, Automatica 40 (4) (2004) 563– 574 https://doi.org/10.1016/j.automatica.2003.11.005.
[22] L. Piroddi, W. Spinelli, An identification algorithm for polynomial NARX models based on simulation error minimization, Int. J. Control 76 (17) (2003) 1767–1781, doi:10.1080/00207170310001635419.
[23] L.D. Re, J. Chapuis, V. Nevistic, Predictive control with embedded feedback linearization for bilinear plants with input constraints, in: Proceedings of the 32nd IEEE Conference on Decision and Control, 1993, pp. 2984–2989vol.4, doi:10.1109/CDC.1993.325747.
[24] S.L. Richter, R.A. Decarlo, Continuation methods: theory and applications, IEEE Trans. Syst. Man Cybern. (4/1983) (1983) 459–464. IEEE
[25] D. Rodriguez-Granrose, A. Jones, H. Loftus, T. Tandeski, W. Heaton, K.T. Foley, L. Silverman, Design of experiment (doe) applied to artificial neural network architecture enables rapid bioprocess improvement, Bioprocess Biosyst. Eng. 44 (6) (2021) 1301–1308.
[26] M. Sassano, A. Astolfi, Dynamic solution of the HJB equation and the optimal control of nonlinear systems, in: Proceeding of the 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 3271–3276, doi:10.1109/CDC.2010.5716990.
[27] M. Sassano, A. Astolfi, Dynamic approximate solutions of the HJ inequality and of the HJB equation for input-affine nonlinear systems, IEEE Trans. Autom. Control 57 (10) (2012) 2490–2503, doi:10.1109/TAC.2012.2186716.
[28] M. Sassano, T.E. Passenbrunner, M. Hirsch, L. del Re, A. Astolfi, Approximate optimal control of the air path of a diesel engine, in: Proceedings of the American Control Conference, 2012, pp. 4204–4209.
[29] P. Schrangl, L. Giarr, L. del Re, Online *k*-step model identification with directional forgetting, in: Proceedings of the European Control Conference (ECC) 2019, 2019a.
[30] P. Schrangl, L. Giarr, F. Reiterer, L. del Re, Online *k*-step PNARX identification for nonlinear engine systems, in: Proceedings of the IEEE Conference on Control Technology and Applications (CCTA), 2019b.
[31] P. Schrangl, P. Tkachenko, L. del Re, Iterative model identification of nonlinear systems of unknown structure: systematic data-based modeling utilizing design of experiments, IEEE Control Syst. Mag. 40 (3) (2020) 26–48.
[32] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modeling in system identification: a unified overview, Automatica 31 (12) (1995) 1691–1724.
[33] H. Sussmann, Nonlinear controllability and optimal control, 2017, doi:10.1201/9780203745625.
[34] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (Methodol.) 58 (1) (1996) 267–288.
[35] S.G. Tzafestas, K.E. Anagnostou, T.G. Pimenides, Stabilizing optimal control of bilinear systems with a generalized cost, Optim. Control Appl. Methods 5 (2) (1984) 111–117, doi:10.1002/oca.4660050204.
[36] H.P. Wynn, The sequential generation of *d*-optimum experimental designs, Ann. Math. Stat. 41 (5) (1970) 1655–1664.