International Journal of Control

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tcon20

# Automated nonlinear feedforward controller identification applied to engine air path output tracking

Alexis Benaitier, Stefan Jakubek, Ferdinand Krainer & Christoph Hametner

**To cite this article:** Alexis Benaitier, Stefan Jakubek, Ferdinand Krainer & Christoph Hametner (2023): Automated nonlinear feedforward controller identification applied to engine air path output tracking, International Journal of Control, DOI: 10.1080/00207179.2023.2227740

**To link to this article:** https://doi.org/10.1080/00207179.2023.2227740



Published online: 27 Jun 2023.

INTERNATIONAL JOURNAL OF CONTROL
https://doi.org/10.1080/00207179.2023.2227740

Taylor & Francis
Taylor & Francis Group



# Automated nonlinear feedforward controller identification applied to engine air path output tracking

Alexis Benaitier [a], Stefan Jakubek[b], Ferdinand Krainer[c] and Christoph Hametner[a]

[a]Christian Doppler Laboratory for Innovative Control and Monitoring of Automotive Powertrain Systems, TU Wien Vienna, Vienna, Austria; [b]Institute of Mechanics and Mechatronics, TU Wien Vienna, Vienna, Austria; [c]AVL list GmbH, Graz, Austria

**ABSTRACT**

This paper introduces a feedforward control method for physical systems that can be described with linear parameter-varying (LPV) models. The proposed feedforward controller structure is consequently derived from a generic LPV representation and is shown to be identifiable directly from noisy measurement data. The identified structure is advantageous for feedforward control, as using a simple least squares algorithm allows to parameterise basis functions representing the required input trajectory to follow a given output trajectory. Also, with the proposed regularisation, the input trajectory remains bounded even when the physical system exhibits non-minimum phase behaviour. Additionally, the proposed controller structure does not possess states but only considers the inputs and outputs signals and their derivatives, leading to a unique physical interpretation of each controller's parameter. Multiple feedforward controllers identified at various operating points can therefore be directly merged to create a parameter-varying controller. A nonlinear and locally non-minimum phase system is considered in this study, i.e. an engine air path, to evaluate the performances of the proposed feedforward strategy. The controller parameters are first identified from noisy measurement data, and then the proposed feedforward controller is implemented with a feedback controller to track the exhaust pressure and $NO_x$ concentration. Using a detailed physical simulation of the engine air path, the proposed feedforward strategy showed encouraging output tracking performances compared to state-of-the-art control methods. The presented feedforward method is shown to be straightforward to identify and calibrate while guaranteeing a contained computational complexity and being applicable to many physical systems thanks to its modularity.

**ARTICLE HISTORY**
Received 29 August 2022
Accepted 9 June 2023

**KEYWORDS**
Nonlinear feedforward; output tracking; predictive control; multivariate engine control

## 1. Introduction

Feedforward control is a classical and efficient method to enhance the performances of a feedback controller (Jean-Francois et al., 2009; Poe & Mokhatab, 2017; Zhang et al., 2022). However, no general method exists to identify a feedforward controller of an arbitrary nonlinear physical system. This paper proposes a controller structure that can be easily identified from measurement data and applied to any physical system that can be modelled as a linear parameter-varying (LPV) multi-input multi-output (MIMO) model. A diesel engine air path is taken as an example throughout this paper as multiple studies already successfully identified LPV MIMO models to capture its dynamics (Euler-Rolle et al., 2021; Kang & Shen, 2017; Ortner & Re, 2007; Zhang et al., 2022).

Physical systems potentially exhibit, without loss of generality, nonlinear dynamics, coupling behaviour and non-minimum phase behaviours (John Hauser & Sastry, 1992; Qiu & Davison, 1993). For many fields of applications, simple rule-based and map-based controllers are still predominantly employed. The necessary calibration efforts and the determination of an appropriate control structure nevertheless limit the resulting performance of such controllers. Different systems or systems configurations usually necessitate distinct control strategies, leading to high development costs and efforts. A modular controller structure with an automated identification from measurement data would therefore be highly beneficial in terms of calibration effort and modularity.

Numerous advanced control methods are currently using a hierarchical control framework. It consists of a first control layer defining setpoints for measured signals to achieve optimality with respect to a given metric, e.g. cost, time, reference tracking, etc. These setpoints, or desired trajectories, are usually map-based as in Plianos and Stobart (2011) or result from a static optimisation using a simplified model of the system (Jiang & Shen, 2019). Hierarchical control frameworks typically comprise a second control layer responsible for controlling the actuators to achieve accurate output setpoints or trajectories tracking.

This second control layer has already been investigated to realise an accurate output tracking of the reference. Initially consisting of a single adaptive feedback controller as in Plianos and Stobart (2011), a recent study emphasises the importance of a feedforward control for accurate transient output tracking (Zhang et al., 2022). Indeed, predictive control methods, i.e. when the output trajectory from the first layer is known beforehand for a given horizon, considerably increase the tracking

**CONTACT** Alexis Benaitier alexis.benaitier@tuwien.ac.at

accuracy if the horizon is long enough (Euler-Rolle et al., 2021; Kang & Shen, 2017). The main bottleneck of model predictive controllers is their inherent computational complexity, given the limited capabilities of control hardware devices especially for mobile applications. Even considering simplified algorithms such as an explicit model predictive controller as in Ortner and Re (2007) or a parametrisation of the input for a reduced dimensionality of a nonlinear predictive controller as proposed in Murilo et al. (2014), more straightforward methods are highly desired regarding a forthcoming hardware implementation.

Feedforward control is an interesting candidate to realise accurate output tracking with computationally limited requirements. Indeed, for a specific simplified model of the system, a nonlinear feedforward controller may be developed based on the inverse characteristics of the model (Hirata et al., 2019). Such controllers benefit from reduced computational complexity but still suffer from calibration efforts and a lack of modularity. To achieve a modular feedforward controller, i.e. reusable for different systems or systems configurations, a black-box local model network (LMN) can be of great interest. Especially if each local model is a linear time-invariant (LTI) model, linear control theory can be used to derive a control law for perfect output tracking.

Necessary conditions for the inversion of an LTI model to achieve perfect output tracking have been formulated by Silverman (1969) at the beginning of the seventies. However, since then, no such conditions for the inversion of a multi-input multi-output (MIMO) nonlinear model have been formulated. As a result, various feedforward methods rely on the differential flatness property introduced by Fliess et al. (1995). Flatness-based control designs are of great interest, especially with the concept of flat input (Waldherr & Zeitz, 2008, 2010). Indeed, a flat input can always be found when the system is observable, and the trajectory of the flat input can be directly known from the desired output trajectory. The physical input can thereafter be recovered from the flat input using a differential parametrisation referred to as a dynamic compensator (Jean-Francois et al., 2009).

The main difficulty when using the concept of flat input with a dynamic compensator is the potential non-minimum phase behaviour of the considered system, i.e. unstable zero dynamics. Indeed, non-minimum phase behaviour can appear in numerous physical systems (John Hauser & Sastry, 1992; Sira-Ramírez & Agrawal, 2004). For a system exhibiting a non-minimum phase behaviour, a dynamic compensator method may create an unbounded control input to realise a perfect output tracking (Isidori, 1995). Nevertheless, perfect output tracking of a non-minimum phase system is still possible with a bounded control input if the controller knows the trajectory beforehand, i.e. with a non-causal controller (Chen & Paden, 1996).

A decomposition-based algorithm can also be employed to identify a feedforward controller (Harris McClamroch & Al-Hiddabi, 1998; Spirito & Marconi, 2022). Splitting the system into a minimum phase system and a non-minimum phase system, the idea is to trivially invert the minimum phase part of the original system while compensating in steady-state conditions for the non-minimum phase part. This method can provide acceptable results, but requires knowledge of control engineering, and cannot be applied in full generality for an arbitrary nonlinear system.

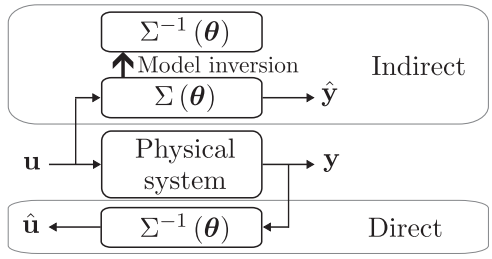


**Figure 1.** Indirect and direct approach for inverse model parameters identification.

In order to avoid difficulties when inverting a model to identify a feedforward controller, a direct identification method can be employed. Schenkendorf and Mangold (2014) indeed proposed two methods to identify the parameters $\boldsymbol{\theta}$ of an inverse model $\Sigma^{-1}$. Classically, an indirect method is used, inverting a model previously identified by fitting a reconstructed output $\hat{\mathbf{y}}$ to the measured output $\mathbf{y}$ given the measured input $\mathbf{u}$, as depicted in Figure 1. Alternatively, a direct method can be considered, where the inverse model is directly identified by fitting the reconstructed input $\hat{\mathbf{u}}$ to the measured input $\mathbf{u}$ given the measured output $\mathbf{y}$, also illustrated in Figure 1. The main advantage of the direct method is that no inversion is necessary, hence no numerical difficulties.

This paper proposes a feedforward controller structure directly identified from measurement data, i.e. direct identification as shown in Figure 1, to ensure robustness against model order selection and applicability to non-minimum phase systems. Additionally, the proposed feedforward controller structure offers the possibility to merge local controllers identified at various operating points to create a single parameter-varying controller. The proposed feedforward method can be applied to a large class of physical systems which can be modelled with an LPV model. The system must be open-loop stable, and the output reference trajectory known and smooth, i.e.sufficiently many times differentiable. Also, the input saturation is not explicitly considered by this method but can be indirectly considered by modifying the output reference, usually limiting the output rate of change. Finally, measurement data have to be available and fulfil persistency of excitation, i.e. all frequencies in the operating range of interest have to be excited.

In this paper, a nonlinear and locally non-minimum phase system is taken as an example; the control of an engine air path. The control inputs are the exhaust gas recirculation valve (EGR) and the variable geometry turbocharger (VGT), controlled to follow a prescribed exhaust manifold pressure $P_{\text{exh}}$ and exhaust nitrogen oxides mass flow $NO_x$ (Murilo et al., 2014; Plianos & Stobart, 2011; Shi & Shen, 2021). Engine air paths are nonlinear systems that are open-loop stable and exhibit strong output coupling (Kang & Shen, 2017; Murilo et al., 2014). This paper mainly focuses on reference output trajectories followable without input saturation. The case where the input is saturated is presented at the end of Section 5, where the output is not perfectly followed, to emphasise the robustness of the proposed method.

The remainder of this paper describes an automated feedforward controller identification method implemented to realise the output tracking of any stable non-linear system. Section 2 details the implementation of the feedforward controller within a classical hierarchical strategy. The proposed control algorithm to generate an input trajectory to track a prescribed output trajectory is presented in Section 3. The identification of the feedforward controller parameters is discussed in Section 4. Finally, the proposed feedforward controller is implemented in combination with a simple feedback controller to track the $NO_x$ and $P_{exh}$ of a diesel engine. Using a detailed simulation platform, the proposed feedforward controller is compared to different classical controllers in Section 5 for fixed and variable engine operating points.

## 2. Control concept

A new feedforward method is proposed in this paper, with a straightforward identification of its parameters and low computational requirements. The controller structure is derived from a generic LPV model; hence it can be used for many physical systems. This section first provides background information regarding feedforward control within a hierarchical control strategy. Then the controller structure is introduced as a transformation of a generic LPV model.

### 2.1 Hierarchichal control strategy

The proposed feedforward control strategy necessitates output reference to be followed and is therefore proposed to be employed in a hierarchical control framework as depicted in Figure 2. The high-level controller generates the desired output trajectory based on the known operating point trajectory of the system. Then a low-level controller designs the required input trajectory to follow the reference output trajectory from the high-level controller. Additionally, the low-level controller can consider feedback from the plant, i.e. the measured physical output, to compensate for model inaccuracies and disturbances.

The reference output trajectory is expected to be smooth in the sense that the reference can be differentiated. Indeed, for many physical systems, the output trajectory essentially consists of smooth transitions between predefined output setpoints and is generated anytime a transition to a new output setpoint is necessary. Additionally, a non-differentiable trajectory, i.e. step, can always be approximated by a smooth trajectory using some filtering techniques. This requirement comes from the fact that most physical systems cannot follow a non-differentiable output reference without an unbounded input unless they exhibit

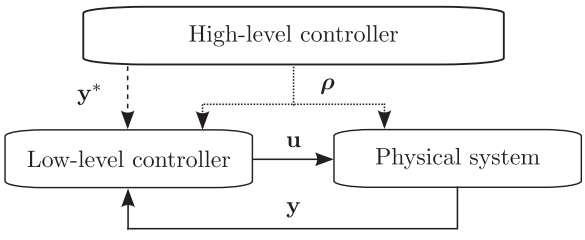


**Figure 2.** Diagram of the low-level controller implementation within the hierarchical control strategy.

a direct feedthrough. To keep the feedforward method general enough, differentiability of the output reference is therefore required.

This paper focuses only on the design of the low-level controller. More specifically, an automated method for feedforward controller identification is proposed and tested. Eventually, a feedback controller is added to the feedforward controller to further study the accuracy and advantages of the proposed feedforward controller. The high-level controller is not considered in this paper, i.e. the desired output trajectory, written with the star superscript $\bullet^*$, is considered perfectly known for the remainder of this paper.

The main assumption of this paper for the design of a feedforward controller is that the physical system can be accurately modelled as an LPV MIMO model for control purposes. The scheduling vector $\boldsymbol{\rho}$, defining the operating point at each instant, is usually taken as the engine speed $N_{ice}$ and the engine torque $T_{ice}$ for an engine air path (Euler-Rolle et al., 2021; Kang & Shen, 2017; Ortner & Re, 2007; Zhang et al., 2022). Without loss of generality, an LPV model can be built as a nonlinear aggregation of LTI models $\Sigma_j$ identified at fixed operating points $\boldsymbol{\rho}_j$

$$\Sigma_j : \begin{cases} \dot{\mathbf{x}}^j = \mathbf{A}^j\mathbf{x}^j + \mathbf{B}^j\mathbf{u} \\ \mathbf{y} = \mathbf{C}^j\mathbf{x}^j + \mathbf{D}^j\mathbf{u} \end{cases}, \tag{1}$$

with $\mathbf{x}^j \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^m, \mathbf{u} \in \mathbb{R}^m$ and the matrices $\mathbf{A}^j, \mathbf{B}^j, \mathbf{C}^j$ and $\mathbf{D}^j$ for each local model. The states $\mathbf{x}^j$ have no physical meaning when the system is identified from black-box identification methods, i.e. when no a-priori knowledge of the system dynamics is known. The states have therefore a different physical meaning at different operating points $\boldsymbol{\rho}_j$ and so different state space parameters cannot be directly merged, i.e. it is not possible to directly interpolate between the matrices $\mathbf{A}^j, \mathbf{B}^j, \mathbf{C}^j$ and $\mathbf{D}^j$. The following section proposes a feedforward controller structure where the controllers' parameters at different operating points can be merged directly.

### 2.2 Feedforward controller structure

A generic feedforward controller structure is proposed in this section, assuming that the physical system can be modelled as an LPV MIMO system. Local controllers are identified at various operating points, with the particularity of all sharing the same parameters' physical interpretation making the design of a parameter-varying controller straightforward.

First, the proposed feedforward structure is introduced at a fixed operating point $\boldsymbol{\rho}_j$, where it is inherited from the linear time-invariant model $\Sigma_j$ defined by the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$, the index $j$ being omitted for the ease of notation. Without loss of generality, $\Sigma_j$ is assumed to be state observable, with the states noted $\mathbf{x}$. The observability matrix can therefore be built blockwise with a relative degree $r_i \geq 1$ associated with each output and fulfilling $\sum_{i=1}^{m} r_i = n$ (Brunovský, 1970)

$$\mathcal{Q} = \begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \\ \cdots \\ \mathcal{Q}_m \end{bmatrix}, \quad \mathcal{Q}_i = \begin{bmatrix} \mathbf{c}_i \\ \mathbf{c}_i\mathbf{A} \\ \vdots \\ \mathbf{c}_i\mathbf{A}^{r_i-1} \end{bmatrix}, \quad \forall i \in \{1, \ldots, m\}, \tag{2a}$$

where $\mathbf{c}_i$ represents the $i$th row of the matrix $\mathbf{C}$.

A linear state transformation using the observability matrix is possible and can be expressed as $\hat{\mathbf{x}} = \mathcal{Q}\mathbf{x}$ leading to the new state space representation

$$\hat{\Sigma}_j : \begin{cases} \dot{\hat{\mathbf{x}}} = \hat{\mathbf{A}}\hat{\mathbf{x}} + \hat{\mathbf{B}}\mathbf{u} \\ \mathbf{y} = \hat{\mathbf{C}}\hat{\mathbf{x}} + \mathbf{D}\mathbf{u} \end{cases}, \tag{3a}$$

$$\hat{\mathbf{A}} = \begin{bmatrix} \hat{\mathbf{A}}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \hline & & \boldsymbol{\Psi}_1^x & & \\ \hline \mathbf{0} & \hat{\mathbf{A}}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \hline & & \boldsymbol{\Psi}_2^x & & \\ \hline & & \vdots & & \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \hat{\mathbf{A}}_m \\ \hline & & \boldsymbol{\Psi}_m^x & & \end{bmatrix}, \tag{3b}$$

with $\boldsymbol{\Psi}_j^x \in \mathbb{R}^{1\times n}, \forall j \in \{1,\dots,m\}$. Each $\boldsymbol{\Psi}_j^x$ is a row vector corresponding to the highest output derivative dynamics of each physical output $j$. The obtained model $\hat{\Sigma}_j$ consists of $m$ chains of integrators $\hat{\mathbf{A}}_i$, one associated with each output. Yet, for an arbitrary system $\Sigma_j$, the inputs can still have a direct impact on all the states $\hat{\mathbf{x}}$, because $\hat{\mathbf{B}}$ and $\mathbf{D}$ have no particular structure. For example, the first $r_1$ states correspond to the first output and verify

$$\hat{x}_k = y_1^{(k-1)} - \mathbf{d}_1\mathbf{u}^{(k-1)} - \sum_{l=1}^{k-1} \hat{\mathbf{b}}_l\mathbf{u}^{(k-l-1)}, \quad \forall k \in \{1,\dots,r_1\}, \tag{4a}$$

with the last state dynamics

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{x}_{r_1} = \boldsymbol{\Psi}_1^x\hat{\mathbf{x}} + \hat{\mathbf{b}}_{r_1}\mathbf{u}. \tag{4b}$$

The states associated with each output can be transformed so that the new states become the outputs and their derivatives. For example, for the first output, using relation (4a), new states $\mathbf{z}$ are defined as

$$z_k = \hat{x}_k + \mathbf{d}_1\mathbf{u}^{(k-1)} + \sum_{l=1}^{k-1} \hat{\mathbf{b}}_l\mathbf{u}^{(k-l-1)}, \quad \forall k \in \{1,\dots,r_1\}, \tag{5}$$

such that $z_k = y_1^{(k-1)}$ holds.

The highest output dynamics can be found by derivating $z_{r_1}$ using (5) and the derivative of $x_k$ given in (4b)

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{z}_{r_1} = \boldsymbol{\Psi}_1^x\hat{\mathbf{x}} + \hat{\mathbf{b}}_{r_1}\mathbf{u} + \mathbf{d}_1\mathbf{u}^{(r_1)} + \sum_{l=1}^{r_1-1} \hat{\mathbf{b}}_l\mathbf{u}^{(r_1-l)}. \tag{6}$$

Finally, $\hat{\mathbf{x}}$ can be expressed as a function of $\mathbf{z}$ and the input and its derivatives in (5) and then replaced in (6) to find

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{z}_{r_1} = \boldsymbol{\Psi}_1^x\mathbf{z} + \sum_{l=0}^{r_1} \Psi_{1,l}^u\mathbf{u}^{(l)}, \tag{7}$$

with $\Psi_{1,l}^u \in \mathbb{R}^{1\times m}$.

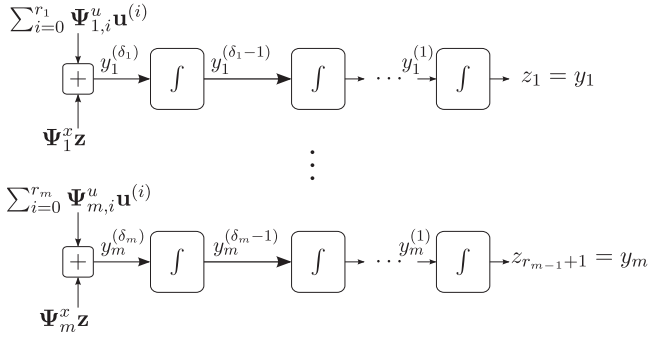


**Figure 3.** Representation of $\tilde{\Sigma}_j$ with $m$ chains of integrators.

This transformation can be done for each output and leads to the new representation presented in Figure 3, where each state corresponds to an output or one of its derivatives

$$\mathbf{z} = \begin{bmatrix} y_1 & y_1^{(1)} & \cdots & y_1^{(r_1-1)} & y_2 & \cdots & y_m^{(r_m-1)} \end{bmatrix}^{\mathrm{T}}. \tag{8}$$

Collecting all the equations representing the highest derivative of each output, i.e. Equation (7) for each output, the following $m$ ordinary differential equations are found

$$\begin{aligned} &\begin{bmatrix} \begin{bmatrix} y_1 \\ y_1^{(1)} \\ \vdots \\ y_1^{(r_1)} \end{bmatrix}^{\mathrm{T}}, & \cdots, & \begin{bmatrix} y_m \\ y_m^{(1)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix}^{\mathrm{T}} \end{bmatrix} \boldsymbol{\theta}_y^j \\ &= \begin{bmatrix} \begin{bmatrix} u_1 \\ u_1^{(1)} \\ \vdots \\ u_1^{(r^*)} \end{bmatrix}^{\mathrm{T}}, & \cdots, & \begin{bmatrix} u_m \\ u_m^{(1)} \\ \vdots \\ u_m^{(r^*)} \end{bmatrix}^{\mathrm{T}} \end{bmatrix} \boldsymbol{\theta}_u^j, \end{aligned} \tag{9}$$

where $r^* \leq \max_i r_i, \forall i \in \{1,\cdots,m\}$ and the feedforward controller parameter matrices $\boldsymbol{\theta}_y^j \in \mathbb{R}^{\sum(r_i+1)\times m}$ and $\boldsymbol{\theta}_u^j \in \mathbb{R}^{m(r^*+1)\times m}$. Each column $k \in \{1,\dots,m\}$ of the matrices $\boldsymbol{\theta}_y^j$ and $\boldsymbol{\theta}_u^j$ are directly built reordering the terms of $\boldsymbol{\Psi}_k^x$ and $\boldsymbol{\Psi}_{k,\cdot}^u$. For the well-conditioned of the feedforward controller, each output's highest dynamics is a weighted sum of the outputs, the inputs and their derivatives and is linearly independent of the other highest output derivatives, i.e. the matrix $\begin{bmatrix} \boldsymbol{\theta}_y^j \\ \boldsymbol{\theta}_u^j \end{bmatrix}$ must be full column rank.

This paper proposes to use the input-output relation (9) as the structure of a feedforward controller. This representation is advantageous as it prevents classical numerical difficulties associated with matrix inversion during parameter identification or integration of unstable dynamics for feedforward control of non-minimum phase systems. Also, this structure does not possess any states; hence no observability problem can appear. The output and input must be smooth enough in the sense that their time derivatives are well defined; this assumption is usually verified for arbitrary physical systems where the input and output cannot physically *jump* but always change smoothly within a

small enough time window. Finally, merging controllers' parameters is straightforward, as each parameter has a unique physical interpretation, independent of the operating point.

In the following of this paper, an efficient method to identify the controller parameters from noisy measurement data is proposed. It is shown to be robust against model order selection and benefits from a numerically efficient total least squares formulation. Also, a simple method only requiring to solve a linear least squares problem to design the necessary inputs to follow a prescribed output trajectory is presented. The proposed feedforward controller is consequently straightforward to calibrate, only consisting of physically interpretable parameters. Furthermore, its computational load is low enough to consider a future hardware implementation.

Section 3 describes the proposed method to use (9) as a feedforward controller, assuming already identified $\boldsymbol{\theta}_y(\boldsymbol{\rho})$ and $\boldsymbol{\theta}_u(\boldsymbol{\rho})$. The identification of the controller parameters $\boldsymbol{\theta}_y(\boldsymbol{\rho})$ and $\boldsymbol{\theta}_u(\boldsymbol{\rho})$ is discussed in Section 4 where first a controller is identified at each operating point, and then a nonlinear aggregation of all the feedforward controllers is built from transient measurements. The performances of the proposed method are evaluated and compared to classical control methods in Section 5 using a high-fidelity simulation platform of a diesel engine air path.

## 3. Input trajectory design

The feedforward controller should provide a trajectory of the inputs $\mathbf{u}$ so that the system outputs $\mathbf{y}$ follow a trajectory prescribed by the high-level controller. This section proposes a robust method to realise such an output tracking using the relation (9) and a parametrisation of the input with basis functions. Regularisation is added to the resulting least squares algorithm to ensure bounded inputs even when the system exhibits non-minimum phase behaviour.

### 3.1 Input parametrisation with basis functions

The output trajectory tracking task can be seen as finding the input trajectory such that the Equation (9) is fulfilled at each time. Assuming that the output reference trajectory and its derivatives are known, and the parameters $\boldsymbol{\theta}_y(\boldsymbol{\rho})$ and $\boldsymbol{\theta}_u(\boldsymbol{\rho})$ identified as in Section 4, a linear system of ordinary differential equations has to be solved to estimate the required input trajectory.

A collocation method is proposed in this paper, as it has already shown successful results for solving ordinary differential equations that are usually difficult to solve with integration methods (Mai-Duy, 2005). Indeed, when the zero dynamics of the system is unstable, i.e. non-minimum phase system, the right-hand side of (9) has at least one unstable eigenmode.

The underlying idea of using a collocation method is to approximate the input infinite-dimensional function space with a finite set of functions

$$u_i = \boldsymbol{\varphi}\boldsymbol{\gamma}_{u_i}, \quad \forall i \in \{1, \dots, m\}, \tag{10}$$

with each parameter vector $\boldsymbol{\gamma}_{u_i} \in \mathbb{R}^L$ and a set of $L$ linearly independent functions

$$\boldsymbol{\varphi} = [\varphi_1, \dots, \varphi_L] \in \mathbb{R}^{1\times L} \tag{11a}$$

$$\varphi_k : \mathbb{R} \to \mathbb{R}, \quad \forall k \in \{1, \dots, L\}. \tag{11b}$$

Gaussian functions are chosen to create a radial basis function network that has been proven to be a universal function approximator (Liao et al., 2003). Also, these functions are infinitely differentiable, straightforward to parameterise and are non-zero only in a small region, impacting the modelled signal only locally.

To create a radial basis function network, Gaussian functions

$$\varphi_k(t) = e^{-\epsilon_k(t-\tau_k)^2}, \tag{12}$$

are concentrated around regularly spaced time locations $\tau_k$.

The parameter $\epsilon_k$ is chosen in a way that neighbouring functions overlap and are sufficiently large to capture local behaviours. By simply plotting the basis functions and the designed inputs, it is straightforward to calibrate $\epsilon_k$ to achieve the desired trade-off between smoothness and accurate tracking. Regarding the number of functions, as regularisation is added in the next section, a large value of $L$ will only increase the computational requirements, while a smaller $L$ will, at one point, deteriorate the tracking accuracy. Choosing $L$ large and decreasing its value until the accuracy is negatively impacted constitutes a simple and efficient calibration method.

For a practical implementation of relation (10), the functions $\varphi_k$ are discretised in $N_\mathrm{t}$ samples. The linear system of ordinary differential Equation (9) can be reformulated as

$$\boldsymbol{\Phi}_u\boldsymbol{\gamma}_u = \boldsymbol{\Phi}_y, \tag{13}$$

with the extended parameter vector $\boldsymbol{\gamma}_u \in \mathbb{R}^{Lm}$ defined in (A4), the matrix $\boldsymbol{\Phi}_u \in \mathbb{R}^{N_\mathrm{t}m\times Lm}$ and the vector $\boldsymbol{\Phi}_y \in \mathbb{R}^{N_\mathrm{t}m}$ as defined in detail in Appendix.

A classical least squares method can be employed to estimate $\boldsymbol{\gamma}_u$ in (13). Nevertheless, a dedicated regularisation is proposed in the following section to keep the inputs bounded.

### 3.2 Bounded input trajectory with regularisation

Some physical systems are challenging to control with feedforward because they exhibit non-minimum phase behaviours. For example, diesel engine air paths usually exhibit non-minimal phase behaviour because of the turbocharger dynamics (Stürzebecher et al., 2015). For any physical system with unstable zero dynamics, a perfect output trajectory tracking can lead to an unbounded control input when the control horizon is bounded in the negative or positive time direction (Chen & Paden, 1996). In that sense, finding a bounded input trajectory so that Equation (9) perfectly holds at each time is usually not possible. Instead, a bounded input trajectory that minimises the error between both sides of Equation (9) at each point in time is proposed. The resulting output tracking will be shown to be close to the expected trajectory, especially when preactuation time is available, and the inputs will remain bounded, i.e. feasible.

This paper proposes to use a modified ridge regression to ensure boundedness of the input trajectory (Ramsay & Silverman, 2005). Taking advantage of the structure of $\boldsymbol{\Phi}_u$ in (13), a penalty directly applied to a specific input and its derivatives is possible.

The coefficients describing the optimal smooth input trajectory $\boldsymbol{\gamma}_u^*$ are directly given as a modified version of the initial Moore–Penrose inverse

$$\boldsymbol{\gamma}_u^* = \left(\boldsymbol{\Phi}_u^{\mathrm{T}}\boldsymbol{\Phi}_u + \mathbf{C}_{\mathrm{reg}}^{\mathrm{T}}\mathbf{C}_{\mathrm{reg}}\right)^{-1}\boldsymbol{\Phi}_u^{\mathrm{T}}\boldsymbol{\Phi}_y, \tag{14a}$$

with $\mathbf{C}_{\mathrm{reg}} \in \mathbb{R}^{N_t m \times Lm}$.

The matrix $\mathbf{C}_{\mathrm{reg}}$ is built as a block diagonal matrix to weight the input and its derivatives independently

$$\mathbf{C}_{\mathrm{reg}} = \mathrm{diag}\left\{\mathbf{C}_{\mathrm{reg},i}\right\}, \quad \forall i \in \{1,\dots,m\}, \tag{14b}$$

where each matrix $\mathbf{C}_{\mathrm{reg},i}$ weights a specific input and its derivatives

$$\mathbf{C}_{\mathrm{reg},i} = \sum_{k=0}^{r^*} \upsilon_{i,k}\boldsymbol{\varphi}^{(k)^{\mathrm{T}}}, \tag{14c}$$

with $\upsilon_{i,k} \geq 0$ the regularisation parameter for the $k$th derivative of the $i$th input and $\boldsymbol{\varphi}^{(k)} \in \mathbb{R}^{N_t \times L}$ corresponding to the sampled $k$th derivatives of the functions defined in (11a). The proposed regularisation plays an essential role in numerical stability to ensure that the least squares problem (13) is well-posed. Also, when the system exhibits non-minimum phase behaviour, regularisation ensures that the inputs remain bounded. In such a case, perfect output tracking is, in theory, only possible with an infinite pre-actuation time (Chen & Paden, 1996; Isidori, 1995). Nevertheless, with enough pre-actuation time, e.g. a few seconds for a diesel engine air path, perfect output tracking can still be realised up to numerical precision. The proposed feedforward controller takes advantage of this possibility; it is, therefore, non-causal, in the sense that the EGR and VGT trajectories are designed in advance for a prescribed horizon.

## 4. Controller parameter identification

The controller parameters $\boldsymbol{\theta}_y(\boldsymbol{\rho})$ and $\boldsymbol{\theta}_u(\boldsymbol{\rho})$ are function of the scheduling variable $\boldsymbol{\rho}$ introduced in Section 2.1. A local learning approach is used to reduce the identification complexity and gain meaningful information from the model structure (Hametner & Jakubek, 2013). First, the local controllers are identified in Section 4.1, and then a nonlinear aggregation of the local controllers is parameterised in Section 4.2 to create a parameter-varying controller.

### 4.1 Local controller parameter identification

The identification of the controller parameters $\boldsymbol{\theta}_y^j$ and $\boldsymbol{\theta}_u^j$ at a fixed operating point $\boldsymbol{\rho}_j$ in (9) corresponds to the identification of the parameters of a system of ordinary differential equations (ODE). Also, because the physical system may exhibit a non-minimum phase behaviour locally, integration methods for the parameter identification may be difficult (Mai-Duy, 2005). A method based on principal differential analysis is therefore chosen to ensure numerical stability and accuracy (Ramsay & Silverman, 2005).

The measured signals during identification, i.e. input and output, are individually modelled with a weighted sum of basis functions, similarly as in (10). The weighted coefficients are calibrated with the available measurements. Consequently, all the necessary time derivatives of the inputs and outputs can be estimated from the basis functions derivatives. The system of ODE (9), for a constant scheduling vector $\boldsymbol{\rho}_j$, can be written as

$$\boldsymbol{\beta}\boldsymbol{\theta}^j = \mathbf{0}, \tag{15}$$

with $\boldsymbol{\theta}^{j^{\mathrm{T}}} = \begin{bmatrix}\boldsymbol{\theta}_u^{j^{\mathrm{T}}} & \boldsymbol{\theta}_y^{j^{\mathrm{T}}}\end{bmatrix}$ the controller parameters to be identified and $\boldsymbol{\beta}$ a matrix being the concatenation of all the required inputs, outputs, and their derivatives as shown in (9).

Because the measured outputs contain noise, and because all the signals are modelled with basis functions, the signals in $\boldsymbol{\beta}$ are only approximations of reality. A total least squares (TLS) method is used to consider these perturbations during the controller parameter identification. The TLS method is used to find the parameters matrix $\boldsymbol{\theta}^j$ that exactly fulfils (15) for a theoretical signals matrix $\bar{\boldsymbol{\beta}}$, assumed with no measurement error and no smoothing approximation. Therefore, the matrix $\boldsymbol{\beta}$ is decomposed into a theoretical *true signals* $\bar{\boldsymbol{\beta}}$ and additive noise $\tilde{\boldsymbol{\beta}}$

$$\boldsymbol{\beta} = \bar{\boldsymbol{\beta}} + \tilde{\boldsymbol{\beta}}. \tag{16}$$

Applying the TLS approach, the signals noise matrix is estimated as the matrix with the minimum Frobenius norm that makes $\bar{\boldsymbol{\beta}}$ $m$-rank deficient

$$\bar{\boldsymbol{\beta}} = \arg\min_{\bar{\boldsymbol{\beta}}}\left\{\left\|\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}\right\|_F\right\}, \tag{17a}$$

$$\mathrm{rank}\left(\bar{\boldsymbol{\beta}}\right) = \mathrm{rank}(\boldsymbol{\beta}) - m. \tag{17b}$$

A solution to this constrained minimisation (17a)–(17b) can be found using the singular value decomposition of the matrix $\boldsymbol{\beta}$

$$\boldsymbol{\beta} = \mathbf{U}\begin{pmatrix}\Sigma_1 & \mathbf{0}\\ \mathbf{0} & \Sigma_2\end{pmatrix}\begin{pmatrix}\mathbf{V}_1^{\mathrm{T}} & \mathbf{V}_2^{\mathrm{T}}\end{pmatrix}^{\mathrm{T}}, \tag{18}$$

where the $m$ smallest singular values are collected in $\Sigma_2$ with the corresponding right singular vectors $\mathbf{V}_2^{\mathrm{T}}$. Removing only the smallest $m$ eigenvalues of $\boldsymbol{\beta}$ would lead to the matrix $\bar{\boldsymbol{\beta}}$ minimising the Frobenius norm (17a) while fulfilling (17b) according to the Eckart–Young–Mirsky theorem (Eckart & Young, 1936). The estimated nullspace of $\boldsymbol{\beta}$ can be directly identified as the remaining part of the singular value decomposition (18), and will be the subspace where each $i$th column of the estimated parameters matrix lies

$$\boldsymbol{\theta}_i^j \in \langle \mathbf{V}_2^{\mathrm{T}} \rangle, \quad \forall i \in \{1,\dots,m\}, \tag{19}$$

and with all columns of $\boldsymbol{\theta}^j$ being linearly independent.

Given that the matrix $(\mathbf{V}_1^{\mathrm{T}} \quad \mathbf{V}_2^{\mathrm{T}})^{\mathrm{T}}$ is orthonormal, the $m$ column vectors of $\mathbf{V}_2^{\mathrm{T}}$ are all orthogonal unit vectors. In that sense, and without loss of generality, taking $\boldsymbol{\theta}^j = \mathbf{V}_2^{\mathrm{T}}$ is a reasonable choice and does not need any specific re-scaling.

The proposed TLS method is only optimal within the assumption of Gaussian noise (Eckart & Young, 1936), yet gives sensibly better results than standard least-squares methods as experienced by the authors. Also, the collected measurement data must persistently excite the system within the whole operating frequency range to accordingly capture the system dynamics.

### 4.2 Local controller network

To create a parameter-varying feedforward controller (9), the parameters of multiple controllers identified at various operating points must be merged. A substantial advantage of the proposed method is that all the local controllers share the same structure. It is, therefore, possible to interpolate between each feedforward model set of parameters to create a local controller network, equivalent to a local *model* network of *controllers* (Hunt & Johansen, 1997). Additionally, local controller models with different output relative degrees can also be merged, adding a zero coefficient to all the missing input and output derivatives.

To capture the nonlinearities of the air path with respect to the engine speed and load, the feedforward controller parameters are defined as a nonlinear aggregation of the parameters of the $N$ local controllers

$$\boldsymbol{\theta}_u(\boldsymbol{\rho}) = \sum_{j=1}^{N} \tilde{\phi}_j(\boldsymbol{\rho})\, \boldsymbol{\theta}_u^j, \tag{20a}$$

$$\boldsymbol{\theta}_y(\boldsymbol{\rho}) = \sum_{j=1}^{N} \tilde{\phi}_j(\boldsymbol{\rho})\, \boldsymbol{\theta}_y^j, \tag{20b}$$

with $\tilde{\phi}_j : \mathbb{R}^2 \to \mathbb{R}$ the validity function associated with the $j$th local model whose parameters are $\boldsymbol{\theta}_u^j$ and $\boldsymbol{\theta}_y^j$. Furthermore, at any operating point, the weighted sum of all the controller parameters is constrained to be unitary to guarantee model consistency and interpretability

$$\sum_{j=1}^{N} \tilde{\phi}_j(\boldsymbol{\rho}) = 1. \tag{21}$$

Gaussian radial basis functions are employed to provide a simple identification while ensuring a modular and interpretable LMN. The validity functions are parameterised as

$$\phi_j(\boldsymbol{\rho}) = e^{\left(-\boldsymbol{\tau}_j^{\mathrm{T}}\left(\boldsymbol{\rho}-\boldsymbol{\rho}_j\right)^2\right)}. \tag{22}$$

The parameters $\boldsymbol{\tau}_j$ are configurable and define the activation range in each scheduling vector dimension. The Gaussian functions (22) could be normalised to meet the requirement (21) without complexifying the optimisation of $\boldsymbol{\tau}_j$. Also, such a normalisation usually suffers from the so-called *reactivation* issue (Anzar & Azeem, 2004); a validity function can be not close to zero in a region far away from its centre $\boldsymbol{\rho}_j$.

To avoid reactivation, the validity function of each local model is forced to reach zero asymptotically outside of a predefined activation region. These activation regions are defined using sigmoid functions in all $\tilde{n}$ directions of the scheduling vector

$$\phi_{\mathrm{act}_j}(\boldsymbol{\rho}) = \prod_{k=1}^{\tilde{n}} \left[ \frac{1}{1+e^{\left(\boldsymbol{\tau}_{\mathrm{act}}^k\left(\boldsymbol{\rho}^k-\boldsymbol{\rho}_j^k-\boldsymbol{\sigma}^k\right)^2\right)}} + \frac{1}{1+e^{\left(\boldsymbol{\tau}_{\mathrm{act}}^k\left(-\boldsymbol{\rho}^k+\boldsymbol{\rho}_j^k-\boldsymbol{\sigma}^k\right)^2\right)}} \right], \tag{23}$$

with $\boldsymbol{\sigma}^k$ a predefined characteristic length scale in the $k$th-direction of the scheduling vector and $\boldsymbol{\tau}_{\mathrm{act}}^k$ the associated transition smoothness parameter in that direction.

For each local model, its validity function used in (20a) and (20b) results from the normalisation of the product of its raw activation function $\phi_j$ and the corresponding activation region function $\phi_{\mathrm{act}_j}$

$$\tilde{\phi}_j(\boldsymbol{\rho}) = \frac{\phi_j(\boldsymbol{\rho})\,\phi_{\mathrm{act}_j}(\boldsymbol{\rho})}{\sum_{k=1}^{N}\left(\phi_k(\boldsymbol{\rho})\,\phi_{\mathrm{act}_k}(\boldsymbol{\rho})\right)}. \tag{24}$$

The validity function parameters $\boldsymbol{\tau}_j$ are directly identified with measurement data for a variable engine operating point. For that purpose, an interior point method is employed to minimise the square difference between the input computed using the current parameter-varying controller and the measured input. The weighting of each local model parameter results in the activation of the $j$th local model only in the surroundings of its identification region, i.e. for $\boldsymbol{\rho}_j$ close to $\boldsymbol{\rho}$ in the sense of the Euclidean norm.

## 5. Simulation results

This section demonstrates the effectiveness of the proposed automated feedforward controller identification method using an experimentally validated simulation platform of a heavy-duty diesel engine (Stefan et al., 2013). All controllers in this section track a randomly generated output trajectory that mimics $P_{\mathrm{exh}}$ and $\mathrm{NO_x}$ concentration under real scenarios. The reference output trajectory is known in advance, and the output measurements are corrupted with realistic noise during parameter identification. The EGR and VGT are considered as input variables and except in the end of Section 5.4, the input is never saturated.

First, the robustness of the local controller identification is discussed in Section 5.1, especially regarding model order selection. Then, Section 5.2 shows the performances of the proposed feedforward method when applied to an operating point where the system exhibits non-minimum phase behaviour. A PI controller is then added to remove steady-state output tracking error in Section 5.3. Finally, in Section 5.4 a parameter-varying feedforward controller is identified and compared to classical control methods to realise accurate output tracking on the entire engine operating region.

### 5.1 Local controller identification

This section emphasises the advantage of directly identifying a feedforward controller, as proposed in Section 4, compared to the inversion of a forward model (1) as detailed in Section 2.2, i.e. indirect identification. For the direct identification, the basis functions (12) are sampled at 50 ms, and spaced every 0.6 s with $\epsilon_k = 0.7$ to balance accuracy and complexity.

When identifying a feedforward controller from the inversion of a forward model, the model order selection greatly influences the feedforward controller accuracy and stability as depicted in Figure 4. For a small model order, the accuracy is

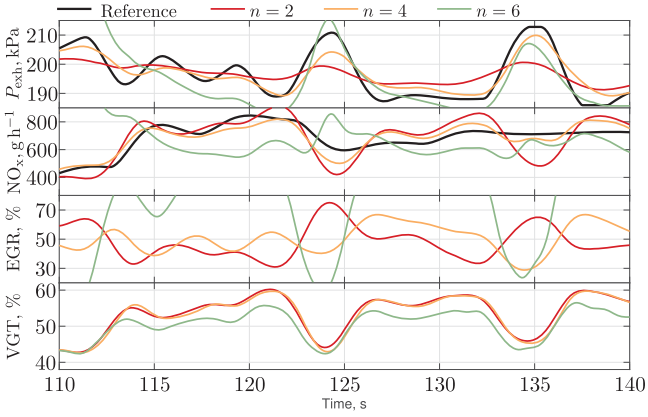


**Figure 4.** Open loop output tracking using different model orders $n$ to identify a feedforward controller from model inversion, for a fixed engine operating point $N_{ice} = 1200$ rpm, $T_{ice} = 1000$ N m.

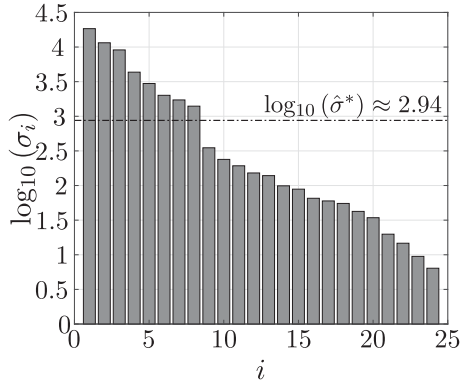


**Figure 6.** Singular values $\sigma_i$ of $\boldsymbol{\beta}$ using up to the 5-th derivative of each input and output. The estimated optimal truncation threshold $\hat{\sigma}^*$ is also represented.

not enough to achieve accurate control. In contrast, some instabilities occur for a high model order because the observability matrix (2a) defined in Section 2.2 is close to being singular.

Directly identifying a feedforward controller from measurement data is more robust to high model order selection, as shown in Figure 5. All the proposed parametrisations in Figure 5 show almost identical results for a sufficiently high model order. When adding additional model parameters, i.e. higher model order, some of these parameters are kept to nearly zero. Another advantage is the ability to independently choose the model order for each output and input. In that sense finding the minimum number of model parameters can be possible, mainly thanks to the TLS method introduced in Section 4.1.

Indeed, with the TLS method and a large number of output and input derivatives, the optimal SVD truncation of (18) can be found as given in Gavish and Donoho (2014). The experimentally computed singular values and the optimal truncation are depicted in Figure 6 using 5th order derivatives for both the output and the input. Ten signals should be kept, as the optimal truncation leads to eight meaningful singular values, and the system has two inputs/outputs according to (17b). A fourth-order model can be chosen, with $r_1 = r_2 = 2$ and $r^* = 1$. This analysis coincides with the discussed results shown in Figure 5.

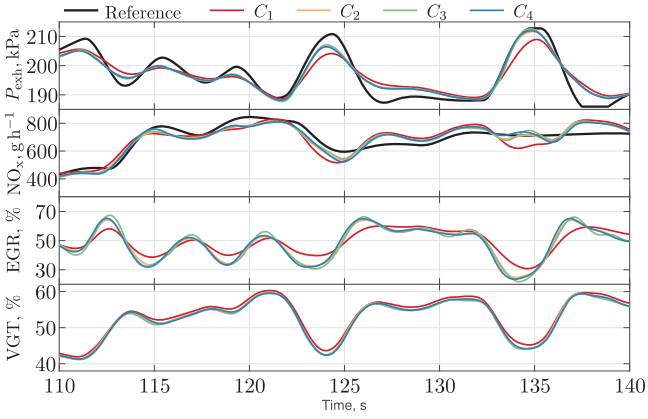


**Figure 5.** Open loop output tracking using different numbers of input/output derivatives to identify a feedforward controller with the TLS method, for a fixed engine operating point $N_{ice} = 1200$ rpm, $T_{ice} = 1000$ N m. ($C_1 : r_1 = r_2 = 1, r^* = 1$; $C_2 : r_1 = r_2 = 2, r^* = 1$; $C_3 : r_1 = r_2 = 2, r^* = 2$; $C_4 : r_1 = r_2 = 3, r^* = 1$).

### 5.2 Feedforward control at a non-minimum phase operating point

A diesel engine air path can exhibit non-minimum phase behaviour when operated at specific operating points, for example, at a high rotational speed of 2000 rpm and a low load of 100 N m, when the VGT is actuated to modify the $NO_x$ mass flow. At this operating point, when the VGT is set proportionally to the desired $NO_x$ concentration, a typical non-minimum-phase behaviour occurs as shown in Figure 7; the output goes in the opposite direction at the beginning of each step.

At this operating point, and for this reduced SISO case, an identified forward model usually has an unstable zero. As a result, when an identified model is inverted to create a feedforward controller, the trajectory designed as proposed in Section 3 necessitates some regularisation to keep the input bounded. Two different levels of regularisation are depicted in Figure 7. Output oscillations and overshoots are reduced with a strong regularisation, but the tracking is still not accurate.

When a feedforward controller is directly identified from measurement data, the input relative degree can be set to a small value to achieve better output tracking accuracy with fewer

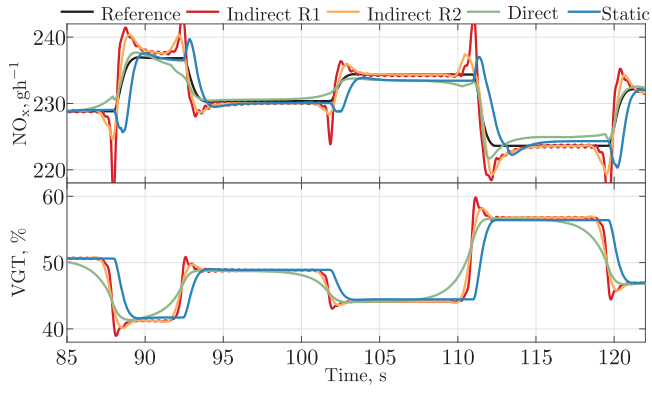


**Figure 7.** Open loop feedforward control of VGT for a fixed operating point $N_{ice} = 2000$ rpm, $T_{ice} = 100$ N m, EGR = 50%, where the system shows non-minimum phase behaviour. (R1: light regularisation, R2: strong regularisation)

oscillations. Indeed, in Figure 7, this latter method, referred to as *direct* identification, shows fewer oscillations and a more accurate output tracking, taking advantage of predictive knowledge regarding the desired output trajectory. With the direct method, regularisation is also needed but is much less sensitive than for the indirect case. Indeed, the regularisation parameters $\upsilon_{i,k}$ in (14c) have been set to 0.1 for each input and each input derivative. Still, noticeably identical results are found whenever the regularisation coefficients are taken in the range $[0.01, 0.5]$.

A *static* controller is also employed in Figure 7 to emphasise the non-minimum phase behaviour of the system at this specific operating point. This controller solely corresponds to a static gain applied to the desired output. At the beginning of each step, the output, i.e. $NO_x$, starts to change in the wrong direction before reaching the desired setpoint. This behaviour occurs because of the non-minimum phase property of the system and is responsible for the numerical difficulties encountered when using a feedforward controller with the indirect method.

### 5.3 Two-degree-of-freedom control

The proposed feedforward controller based on Equation (9) only constitutes an open-loop controller. As a result, steady-state errors are not compensated. Therefore, a PI controller is added to work along with the feedforward controller $C_2$ presented in Figure 5. The PI gains are manually calibrated to asymptotically reach the reference during steady states, with a unique parametrisation for the entire engine operating region. The PI calibration is kept very simplistic, as the PI feedback controller only aims at slowly removing steady-state error.

A schematic representation of this two-degree-of-freedom controller (2DoF) is presented in Figure 8. For this study, the output measurements are considered without noise to emphasise only the feedforward performances. The resulting output tracking of this 2DoF strategy is depicted in Figure 9. The results using only the feedforward controller are also displayed to emphasise the importance of the feedforward part compared to the PI contribution.

### 5.4 Varying engine operating point

In this section, multiple local controllers identified at various engine operating points are merged to create a single parameter-varying controller. After adding a PI feedback loop to this parameter-varying controller, the resulting 2DoF controller is compared to classical controllers for a varying engine operating point.

After identifying local controllers, the validity functions parameters of the parameter-varying controller are optimised

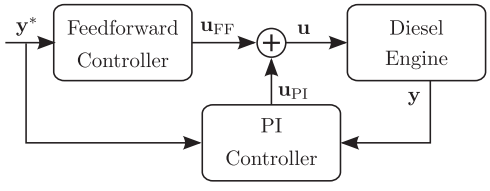


**Figure 8.** Diagram of the two-degree-of-freedom controller (2DoF).

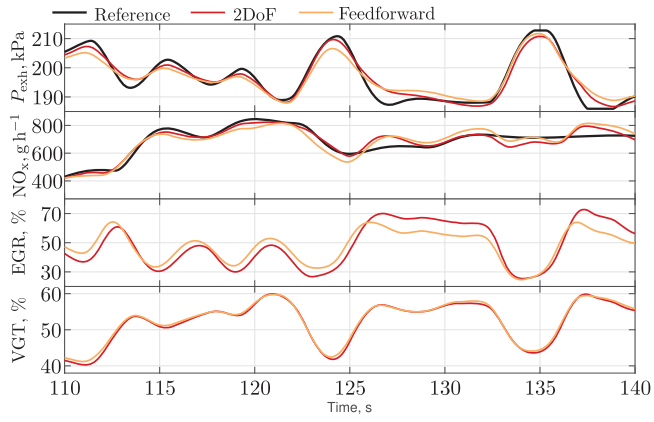


**Figure 9.** Comparison between a feedforward and a 2DoF controller, for a fixed engine operating point $N_{ice} = 1200$ rpm, $T_{ice} = 1000$ N m.

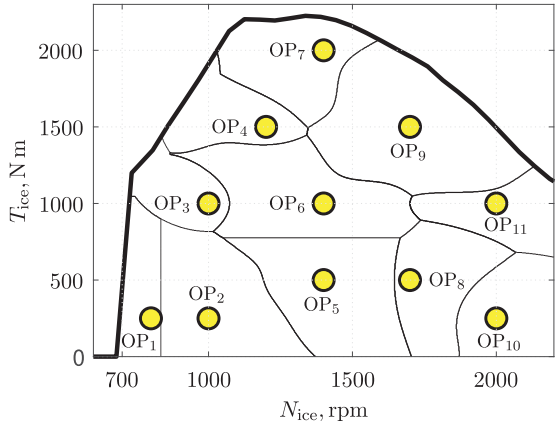


**Figure 10.** Region of highest activation for each local model ($OP_j$) around its identification centre.

with an interior-point method using transient operating point measurement data. Each local model has the highest validity function of them all around its operating point of identification, as shown in Figure 10. The use of regularisation and region of activation, as defined in (24), impedes reactivation and ensures that each local model has a validity function near unity around its corresponding operating point.

The parameter-varying feedforward controller is implemented with a PI controller calibrated with constant gains. This resulting 2DoF controller performances are shown in Figure 11 for a varying engine operating point and compared to:

- Only the PI controller from the 2DoF strategy. This strategy is neither adaptive nor predictive but is highly convenient regarding calibration effort and computational requirements;
- A network of full-state feedback controllers with integration of the control error. It is composed of multiple state feedback controllers identified at various operating points (Gregorcic & Lightbody, 2010). This control method is adaptive but not predictive, as it only uses the current desired output and system measurements;
- A flatness-based MPC as proposed in Euler-Rolle et al. (2021) so that the local model parameters can be

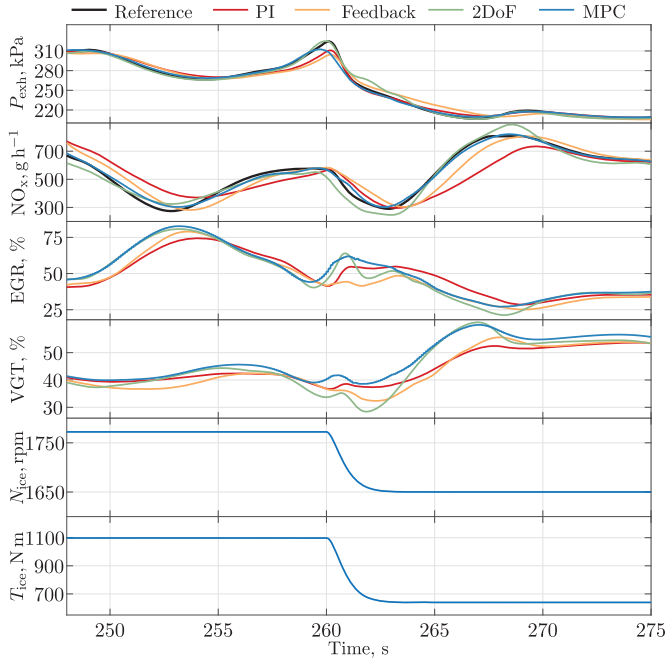


**Figure 11.** Closed-loop results using different controllers, without input saturation and for a varying engine operating point.

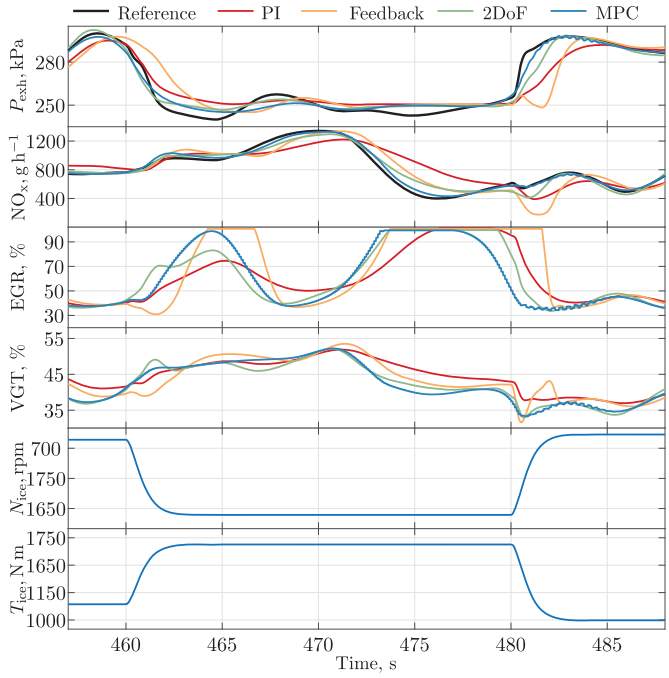


**Figure 12.** Closed-loop results using different controllers, with input saturation and for a varying engine operating point.

directly merged, creating a time-varying controller at each iteration. This adaptive and predictive strategy requires a significant computational effort, making its implementation on current hardware impractical.

The reference tracking regarding the first output, the exhaust pressure $P_{exh}$, is almost identical for all controllers. Only the non-predictive controllers, i.e. PI and state feedback controllers, are slightly less accurate before large pressure variations. Regarding the second output, the $NO_x$ mass flow, output tracking is inaccurate for the PI controller during transient phases. The state feedback controller is faster than the PI controller but fails to accurately follow the desired reference $NO_x$ signal during large transients. These observations are confirmed by a lower coefficient of determination detailed in Table 1 for the non-predictive controllers, especially regarding the second output $NO_x$.

The proposed 2DoF controller is more precise for both outputs during transient phases as it inherently considers predictive information. The 2DoF method is almost as accurate as the MPC and can still be improved with a more sophisticated feedback loop. Indeed, the MPC is primarily relying on its feedback information to control the outputs during transient engine operating points. The 2DoF, with its simple PI feedback loop, cannot perfectly follow the outputs during transient operating points,

**Table 1.** Coefficients of determination for different controllers proposed in Figure 11.

| Controller | $R^2$-$P_{exh}$ | $R^2$-$NO_x$ |
|---|---|---|
| PI | 0.984 | 0.613 |
| Feedback | 0.963 | 0.857 |
| 2DoF | 0.991 | 0.947 |
| MPC | 0.992 | 0.987 |

**Table 2.** Coefficients of determination for different controllers proposed in Figure 12.

| Controller | $R^2$-$P_{exh}$ | $R^2$-$NO_x$ |
|---|---|---|
| PI | 0.837 | 0.600 |
| Feedback | 0.676 | 0.709 |
| 2DoF | 0.946 | 0.922 |
| MPC | 0.965 | 0.984 |

but the control error is immediately corrected after the engine operating point returns to a steady state.

Another reference output trajectory tracking is presented in Figure 12 for the same controllers. Nevertheless, this time, the input is saturated for all the controllers as the reference is not reachable between 475 and 480 seconds. Especially the exhaust pressure cannot be accurately followed, even with the adaptive MPC, resulting in lower coefficients of determination given in Table 2 compared to the previous case summarised in Table 1.

The same observation as in the previous case still holds, with non-predictive methods being slower and, therefore, less accurate. During and shortly after an input saturation, the 2DoF method is almost as precise as the adaptive MPC but with significantly smaller computation requirements. Also, compared to the PI or the state feedback controllers, the 2DoF method is much faster at removing a steady-state error after an input saturation phase.

## 6. Conclusion and outlook

This paper proposes an automated method for identifying a feedforward controller for output tracking of a nonlinear physical system. The proposed structure benefits from its modularity, making it applicable to any physical system modelled with an LPV model. Additionally, the controller parameters

can be directly identified from measurement data, guaranteeing robustness against model order selection and ensuring numerical stability. Also, with the suggested TLS identification approach, the model order can be estimated without prior knowledge regarding the physical system.

A significant advantage of the proposed feedforward identification method is that multiple local controllers can be merged. Because they share an identical structure, their parameters can be combined to create a local controller network. Using a single least squares algorithm, the resulting parameter-varying controller can design an entire input trajectory given a desired output trajectory. Moreover, the input trajectory is guaranteed to be smooth and bounded thanks to regularisation, even if the system exhibits non-minimum phase behaviour.

The proposed feedforward controller, implemented with a simple PI feedback loop, i.e. two-degree-of-freedom controller, is compared to classical controllers using a detailed physical simulation of an engine air path. The accuracy during transient output tracking is improved using the proposed strategy compared to non-predictive methods. The 2DoF controller performs almost as well as an adaptive MPC, even with input constraints, but uses only a fraction of its complexity. The input trajectory can therefore be easily recomputed anytime a new and more accurate operating point trajectory or output reference trajectory is available.

Future work is focusing on communicating the generated input trajectory to the high-level controller to improve the setpoints definition and assess input constraints. Additionally, the practical implementation of the proposed controller, e.g. on a diesel engine testbed, is the next step to validate the proposed feedforward strategy. Finally, a theoretical study regarding the optimality of the proposed feedforward controller is also under consideration.

## Acknowledgments

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, the Christian Doppler Research Association as well as AVL List GmbH, Austria is gratefully acknowledged.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## ORCID

*Alexis Benaitier* http://orcid.org/0000-0001-7946-1640

## References

Anzar, M., & Azeem, M. F. (2004). Effect of normalization of membership functions in premise region of generalized adaptive neuro-fuzzy inference system. In *Proceedings of the IEEE INDICON 2004. First India annual conference* (pp. 294–298). IEEE.

Brunovský, P. (1970). A classification of linear controllable systems. *Kybernetika*, *6*(3), 173–188. http://eudml.org/doc/28376

Chen, D., & Paden, B. (1996). Stable inversion of nonlinear non-minimum phase systems. *International Journal of Control*, *64*(1), 81–97. https://doi.org/10.1080/00207179608921618

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, *1*(3), 211–218. https://doi.org/10.1007/BF02288367

Euler-Rolle, N., Krainer, F., Jakubek, S., & Hametner, C. (2021). Automated synthesis of a local model network based nonlinear model predictive controller applied to the engine air path. *Control Engineering Practice*, *110*(October 2019), Article 104768. https://doi.org/10.1016/j.conengprac.2021.104768

Fliess, M., Levine, J., Martin, P., & Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, *61*(6), 1327–1361. https://doi.org/10.1080/00207179508921959

Gavish, M., & Donoho, D. L. (2014). The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, *60*(8), 5040–5053. https://doi.org/10.1109/TIT.2014.2323359

Gregorcic, G., & Lightbody, G. (2010). Nonlinear model-based control of highly nonlinear processes. *Computers & Chemical Engineering*, *34*, 1268–1281. https://doi.org/10.1016/j.compchemeng.2010.03.011

Hametner, C., & Jakubek, S. (2013). Local model network identification for online engine modelling. *Information Sciences*, *220*, 210–225. Online Fuzzy Machine Learning and Data Mining, https://www.sciencedirect.com/science/article/pii/S0020025512000138

Harris McClamroch, N., & Al-Hiddabi, S. A. (1998). A decomposition based control design approach to output tracking for multivariable nonlinear non-minimum phase systems. In *Proceedings of the 1998 IEEE international conference on control applications (Cat. No.98CH36104)* (Vol. 2, pp. 1414–1418). IEEE.

Hirata, M., Hayashi, T., Takahashi, M., Yamasaki, Y., & Kaneko, S. (2019). A nonlinear feedforward controller design taking account of dynamics of turbocharger and manifolds for diesel engine air-Path system. *IFAC-PapersOnLine*, *52*(5), 341–346. 9th IFAC Symposium on Advances in Automotive Control AAC 2019, https://www.sciencedirect.com/science/article/pii/S2405896319306767

Hunt, K. J., & Johansen, T. A. (1997). Design and analysis of gain-scheduled control using local controller networks. *International Journal of Control*, *66*(5), 619–652. https://doi.org/10.1080/002071797224487

Isidori, A. (1995). *Nonlinear control systems* (3rd ed.). Springer London.

Jean-Francois, S., Ferdinand, S., & Kennel, R. (2009). Trajectory tracking control with flat inputs and a dynamic compensator. In *Proceedings of the european control conference* (pp. 248–253). IEEE.

Jiang, W., & Shen, T. (2019). Lyapunov-based nonlinear feedback control design for exhaust gas recirculation loop of gasoline engines. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, *141*(5), 1–11. https://doi.org/10.1115/1.4042146

John Hauser, G. M., & Sastry, S. (1992). Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft. *Automatica*, *28*(4), 665–679. https://www.sciencedirect.com/science/article/pii/000510989290029F

Kang, M., & Shen, T. (2017). Experimental comparisons between LQR and MPC for spark-ignition engine control problem. In *Chinese Control Conference, CCC* (pp. 2651–2656). IEEE.

Liao, Y., Fang, S.-C., & Nuttle, H. L. W. (2003). Relaxed conditions for radial-basis function networks to be universal approximators. *Neural Networks*, *16*(7), 1019–1028. https://www.sciencedirect.com/science/article/pii/S0893608002002277

Mai-Duy, N. (2005). Solving high order ordinary differential equations with radial basis function networks. *International Journal for Numerical Methods in Engineering*, *62*, 824–852. https://doi.org/10.1002/(ISSN)1097-0207

Murilo, A., Alamir, M., & Alberer, D. (2014). A general NMPC framework for a diesel engine air path. *International Journal of Control*, *87*(10), 2194–2207. https://doi.org/10.1080/00207179.2014.905708

Ortner, P., & Re, L. (2007). Predictive control of a diesel engine air path. *IEEE Transactions On Control Systems Technology*, *32*(2), 449–456. https://doi.org/10.1109/TCST.2007.894638

Plianos, A., & Stobart, R. K. (2011). Nonlinear airpath control of modern diesel powertrains: A fuzzy systems approach. *International Journal of Systems Science*, *42*(2), 263–275. https://doi.org/10.1080/00207721.2010.521864

Poe, W. A., & Mokhatab, S. (2017). Chapter 3–process control. In W. A. Poe & S. Mokhatab (Eds.), *Modeling, control, and optimization of natural gas processing plants* (pp. 97–172). Gulf Professional Publishing.

https://www.sciencedirect.com/science/article/pii/B9780128029619000036
Qiu, L., & Davison, E. J. (1993). Performance limitations of non-minimum phase systems in the servomechanism problem. *Automatica*, *29*(2), 337–349. https://www.sciencedirect.com/science/article/pii/000510989390127F
Ramsay, J., & Silverman, B. W. (2005). *Functional data analysis*. Springer-Verlag New York.
Schenkendorf, R., & Mangold, M. (2014). Parameter identification for ordinary and delay differential equations by using flat inputs. *Theoretical Foundations of Chemical Engineering*, *48*(5), 594–607. https://doi.org/10.1134/S0040579514050224
Shi, H., & Shen, T. (2021). Design and experimental validation for nonlinear control of internal combustion engines with EGR and VVT. *SICE Journal of Control, Measurement, and System Integration*, *14*(1), 51–58. https://doi.org/10.1080/18824889.2021.1874680
Silverman, L. (1969). Inversion of multivariable linear systems. *IEEE Transactions on Automatic Control*, *14*(3), 270–276. https://doi.org/10.1109/TAC.1969.1099169
Sira-Ramírez, H., & Agrawal, S. K. (2004). *Differentially flat systems* (1st ed.). CRC Press.
Spirito, M., & Marconi, L. (2022). Inner-outer decomposition for strictly proper linear time invariant systems and non-minimum phase performance limitations. In *2022 IEEE 61st conference on decision and control (CDC)* (pp. 5492–5497). IEEE.
Stefan, P., Michael, K., Efstratios, F., & Atzler, H. (2013). Model-based production calibration–the efficient way of variant development. *MTZ Worldwide*, *74*(10), 18–25. https://doi.org/10.1007/s38313-013-0096-2
Stürzebecher, C., Reß, J., Bohn, C., Märzke, F., & Prase, R. (2015). A diesel engine model including exhaust flap, intake throttle, LP-EGR and VGT. Part II: Identification and validation. *IFAC-PapersOnLine*, *28*(15), 60–65. http://dx.doi.org/10.1016/j.ifacol.2015.10.009
Waldherr, S., & Zeitz, M. (2008). Conditions for the existence of a flat input. *International Journal of Control*, *81*(3), 439–443. https://doi.org/10.1080/00207170701561443
Waldherr, S., & Zeitz, M. (2010). Flat inputs in the MIMO case. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *43*(14), 695–700. https://doi.org/10.3182/20100901-3-IT-2016.00147
Zhang, J., Amini, M. R., Kolmanovsky, I., Tsutsumi, M., & Nakada, H. (2022). Benefits of feedforward for model predictive airpath control of diesel engines. In *10th IFAC symposium on robust control design*. Elsevier.

## Appendix. Matrices for the feedforward least squares formulation

This appendix provides details for re-writing (9) into the form (13) where the input parametrization $\boldsymbol{\gamma}_u$ appears linearly. First, the desired outputs signals are discretised in $N_t$ samples, and so are the basis functions given in (11a), i.e. $\varphi_k \in \mathbb{R}^{N_t}$ leading to $\mathbf{y}_i \in \mathbb{R}^{N_t}, \forall i \in \{1,\dots,m\}$. Additionally, the parameters of the controller structure (9) are also discretised

$$\bar{\boldsymbol{\theta}}_y = \boldsymbol{\theta}_y\left(\boldsymbol{\rho}\left(t\right)\right), \tag{A1}$$

with $\boldsymbol{\rho}(t)$ the discretised operating trajectory provided by the high-level controller. The model parameters are ordered in a three dimensional matrix, i.e. $\bar{\boldsymbol{\theta}}_y \in \mathbb{R}^{n\times m\times N_t}$.

With the discretised expected outputs and their derivatives, the left-hand side of (9) is trivially computed. Also, the resulting $m$ columns of $N_t$ samples are stored in a column vector $\boldsymbol{\Phi}_y$ of $N_t m$ elements

$$\boldsymbol{\Phi}_y = \begin{bmatrix} \sum_{k=1}^{m}\left(\sum_{l=0}^{r_k}\left(y_k^l \odot \bar{\boldsymbol{\theta}}_y\left(l+1+\sum_{i=1}^{k-1} r_i+1,1,\cdot\right)\right)\right) \\ \vdots \\ \sum_{k=1}^{m}\left(\sum_{l=0}^{r_k}\left(y_k^l \odot \bar{\boldsymbol{\theta}}_y\left(l+1+\sum_{i=1}^{k-1} r_i+1,m,\cdot\right)\right)\right) \end{bmatrix}, \tag{A2}$$

with $\bar{\boldsymbol{\theta}}_y(l+1+\sum_{i=1}^{k-1} r_i+1,k,\cdot) \in \mathbb{R}^{N_t}, \forall l \in \{0,\dots,r_k\}, \forall k \in \{1,\dots,m\}$. The notation $\odot$ refers to the standard Hadamard product, resulting in the column vector $\boldsymbol{\Phi}_y \in \mathbb{R}^{N_t m}$.

The right-hand side of (9) is discretised in $N_t$ samples, with $\bar{\boldsymbol{\theta}}_u \in \mathbb{R}^{m(r^*+1)\times m\times N_t}$ created identically as $\bar{\boldsymbol{\theta}}_y$. The inputs are parameterised with basis functions as shown in (10). It follows that in the right-hand side of (9) all the inputs and their derivatives can be written

$$\left[\bar{\mathbf{u}}_1,\quad \bar{\mathbf{u}}_1^{(1)},\quad \dots,\quad \bar{\mathbf{u}}_m^{(r^*)}\right] = \left[\boldsymbol{\varphi}\boldsymbol{\gamma}_{u_1},\quad \boldsymbol{\varphi}^{(1)}\boldsymbol{\gamma}_{u_1},\quad \dots,\quad \boldsymbol{\varphi}^{(r^*)}\boldsymbol{\gamma}_{u_m}\right]. \tag{A3}$$

In Equation (9), the right-hand side can be expressed as done for the left-hand side (A2), just changing the output signals with the inputs and $\bar{\boldsymbol{\theta}}_y$ by $\bar{\boldsymbol{\theta}}_u$. Using the relation (A3), and collecting all the unknown variables,

$$\boldsymbol{\gamma}_u = \left[\boldsymbol{\gamma}_{u_1}^{\mathrm{T}},\quad \boldsymbol{\gamma}_{u_2}^{\mathrm{T}},\quad \dots,\quad \boldsymbol{\gamma}_{u_m}^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{A4}$$

the right-hand side of (9) becomes

$$\boldsymbol{\Phi}_u = \begin{bmatrix} \sum_{k=1}^{m}\left(\sum_{l=0}^{r^*}\left(\boldsymbol{\varphi}^l \odot \bar{\boldsymbol{\theta}}_u\left(l+1+(k-1)\left(r^*+1\right),1,\cdot\right)\right)\gamma_{u_k}\right) \\ \vdots \\ \sum_{k=1}^{m}\left(\sum_{l=0}^{r^*}\left(\boldsymbol{\varphi}^l \odot \bar{\boldsymbol{\theta}}_u\left(l+1+(k-1)\left(r^*+1\right),m,\cdot\right)\right)\gamma_{u_k}\right) \end{bmatrix}. \tag{A5}$$

From (A5), all contributions $\gamma_{u_k}, \forall k \in \{1,\dots,m\}$ can be collected to become a linear operation with respect to $\boldsymbol{\gamma}_u$

$$\boldsymbol{\Phi}_u\boldsymbol{\gamma}_u = \begin{bmatrix} \begin{bmatrix} \sum_{l=0}^{r^*}\left(\boldsymbol{\varphi}^{(l)} \odot \bar{\boldsymbol{\theta}}_u\left(l+1+(1-1)\left(r^*+1\right),1,\cdot\right)\right) \\ \vdots \\ \sum_{l=0}^{r^*}\left(\boldsymbol{\varphi}^{(l)} \odot \bar{\boldsymbol{\theta}}_u\left(l+1+(m-1)\left(r^*+1\right),1,\cdot\right)\right) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \sum_{l=0}^{r^*}\left(\boldsymbol{\varphi}^{(l)} \odot \bar{\boldsymbol{\theta}}_u\left(l+1+(1-1)\left(r^*+1\right),m,\cdot\right)\right) \\ \vdots \\ \sum_{l=0}^{r^*}\left(\boldsymbol{\varphi}^{(l)} \odot \bar{\boldsymbol{\theta}}_u\left(l+1+(m-1)\left(r^*+1\right),m,\cdot\right)\right) \end{bmatrix} \end{bmatrix} \boldsymbol{\gamma}_u, \tag{A6}$$

with $\boldsymbol{\Phi}_u \in \mathbb{R}^{N_t m\times Lm}$.