2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

LPV State-Space Identification via IO Methods and Efficient Model Order Reduction in Comparison with Subspace Methods

Erik Schulz, Pepijn B. Cox, Roland Tóth and Herbert Werner

***Abstract*— In this paper, we introduce a procedure for global identification of linear parameter-varying (LPV) discrete-time state-space (SS) models with a static, affine dependency structure in a computationally efficient way. The aim is to develop off-the-shelf LPV-SS estimation methods to make identification practically accessible. The benefits of identifying a computational straightforward LPV input-output (IO) model – that has an equivalent SS representation with static, affine dependency – is combined with an LPV-SS model order reduction. To increase practical relevance of the proposed scheme, in this paper, we present a computational attractive model order reduction scheme based on the LPV Ho-Kalman like realization scheme. We analyze the computational complexity and scalability of our method and compare its benefits to the PBSID$_{\text{opt}}$ scheme. Two examples are provided to demonstrate that our introduced approach performs similar to PBSID$_{\text{opt}}$ in a numerical example and outperforms the PBSID$_{\text{opt}}$ on measurements of a real world system, the air-path system of a gasoline engine.**

I. INTRODUCTION

The LPV representation provides a framework to embed a large variety of nonlinear and time-varying phenomena of the underlying system. Accordingly, the LPV framework makes systematic model based control design with non-stationary and/or time-varying behavior possible, due to its roots in the linear time-invariant (LTI) theory. Thus, in the last two decades, the theory of LPV systems has developed considerably [1] and LPV control has been applied to a wide range of applications, e. g., in aerospace [2], automotive industry [3] or robotics [4]. See [1] for an overview.

In general, synthesizing an LPV controller is based on an LPV model in SS form with static, affine dependency on the scheduling signal. Such a required model can be obtained by identifying a global LPV model from measurements of the system. Two classes of approaches for identification of LPV systems are considered here, which have received attention in the literature. The first is the identification of LPV-IO models introduced in [5], [6], [7]. However, they need to be transformed into a state-space form, where general IO to SS conversion introduces dynamic and rational dependencies on the scheduling signal and/or results in non-minimal state realization [7]. To attain an SS realization with static, affine dependency, three different IO forms with specific dynamic dependency structure were considered in [8], [9], which have a direct SS realization. In the second class of approach, this problem is avoided altogether using LPV subspace identification (SID) methods. Here a state-space representation with an assumed dependency structure is directly obtained [10], [11], [12], however, SID methods have a significant higher computational load.

In this paper, we are interested in identifying LPV-SS models with static and affine dependency in a computational efficient way. Yet, the low complexity LPV-IO identification method with a direct SS realization will generally result in models with non-minimal state dimension, especially in the multi-input multi-output (MIMO) case [8], [9]. Therefore, as part of the identification procedure, a fast LPV model order reduction scheme based on the Ho-Kalman like algorithm [8] is presented in this paper. The computational load of the original reduction scheme could be reduced tremendously as an SS realization of the LPV model is available a-priori. As an example will show, the resulting reduced LPV-SS models attain similar prediction accuracy for equivalent state order as the models from SID. Advantageously, the proposed scheme scales significantly better compared to SID in a computational sense, which increases attraction of this LPV-IO identification even for moderate or large scale systems, where subspace methods are not applicable due to their computational complexity.

This paper is organized as follows: in Section II, the LPV-SS model is introduced. Then, in Section III, the LPV-IO and LPV-SID methods are presented together with differences and similarities of both approaches. In Section IV, a fast, computational attractive model order reduction scheme is given. Next, in Section V two case studies are presented followed by the conclusions in Section VI.

II. THE LPV STATE-SPACE MODEL

A discrete-time LPV model in state-space form has the following representation

$$G(\theta)=\left[\begin{array}{c|c}A(\theta) & B(\theta)\\ \hline C(\theta) & D(\theta)\end{array}\right]:\begin{cases}x_{k+1}=A(\theta_k)x_k+B(\theta_k)u_k\\ \quad y_k=C(\theta_k)x_k+D(\theta_k)u_k\end{cases}$$

where $x_k \in \mathbb{R}^{n_x}$, $y_k \in \mathbb{R}^{n_y}$ and $u_k \in \mathbb{R}^{n_u}$ are the state, output and input vectors, respectively. For brevity, the time dependence is expressed by the index, e. g., $y(k) = y_k$, or omitted completely if it is clear from the context. The scheduling parameter vector $\theta_k \in \mathcal{P} \subseteq \mathbb{R}^{n_\theta}$ is time-varying and online measurable where $\mathcal{P}$ is a compact set. The output data collected can obtain additive colored noise, but, for the sake of simplicity, the input and scheduling signals are assumed to be measured noise free. The model matrices

This work was supported by IAV GmbH.

Erik Schulz is with IAV GmbH, 38518 Gifhorn, Germany erik.schulz@iav.de

Pepijn B. Cox and Roland Tóth are with the Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB, The Netherlands {p.b.cox, r.toth}@tue.nl

Herbert Werner is with the Institute of Control Systems, Hamburg University of Technology, 21073 Hamburg, Germany www.tuhh.de/rts

978-1-5090-2872-6/17/$31.00 ©2017 IEEE

$A(\theta) \in \mathbb{R}^{n_x \times n_x}$, $B(\theta) \in \mathbb{R}^{n_x \times n_u}$, $C(\theta) \in \mathbb{R}^{n_y \times n_x}$ and $D(\theta) \in \mathbb{R}^{n_y \times n_u}$ are the scheduling parameter-dependent system, input, output and direct feedthrough matrices, respectively. In this paper only affine parameter dependency is considered, e. g., one can express the system matrices as:

$$A(\theta_k) = A(\mu_k) = \sum_{l=1}^{n_\mu} \mu_k^{(l)} A^{(l)},$$

where $\mu_k^\top = \begin{bmatrix} 1 & \theta_k^\top \end{bmatrix}$ is also denoted as scheduling parameter vector and the superscript (l) indicates the l-th entry of the scheduling vector μ_k and local model matrix $A^{(l)} \in \mathbb{R}^{n_x \times n_x}$.

III. LPV IDENTIFICATION METHODS

In general, LPV control synthesis methods require an LPV-SS model with static and affine dependency on the scheduling signal, i. e., require $G(\theta)$. However, current state-of-the-art subspace identification schemes have a significant computational load [10, Algorithms 1 and 2]. To this end, we introduce a simplified identification scheme based on a low complexity IO structure that has a direct realization in terms of a static, affine SS form. We will highlight the computational benefits and stochastic efficiency of this approach later.

A. LPV-IO Identification with SS Realization

For IO identification, the data-generating system is assumed to be in the following autoregressive moving average with exogenous input (ARMAX) form:

$$A(q, \theta_k) y_k = B(q, \theta_k) u_k + C(q, \theta_k) e_k, \tag{1}$$

which can be equivalently represented as

$$\begin{aligned} y_k &= G(q, \theta_k) u_k + H(q, \theta_k) e_k \\ &= A^\dagger(q, \theta_k) B(q, \theta_k) u_k + A^\dagger(q, \theta_k) C(q, \theta_k) e_k, \end{aligned} \tag{2}$$

if the left inverse $A^\dagger(q, \theta_k)$ of $A(q, \theta_k)$ exists, i. e., $A^\dagger(q, \theta_k) A(q, \theta_k) = I$ in the functional sense[1]. In (2), $G(q, \theta_k)$ is referred to as the system model and $H(q, \theta_k)$ as the noise model. The matrix polynomials $A(q, \theta_k)$, $B(q, \theta_k)$, $C(q, \theta_k)$ in (1) – (2) are given as

$$A(q, \theta_k) = I + \sum_{i=1}^{n_a} a_i(q^{-i}\theta_k) q^{-i}, \tag{3a}$$

$$B(q, \theta_k) = b_0^{(1)} + \sum_{j=1}^{n_b} b_j(q^{-j}\theta_k) q^{-j}, \text{ and} \tag{3b}$$

$$C(q, \theta_k) = I + \sum_{m=1}^{n_c} c_j(q^{-m}\theta_k) q^{-m}. \tag{3c}$$

The noise vector $e_k \in \mathbb{R}^{n_y}$ is assumed to be zero mean white and independent of past input, scheduling, and output data, q is the forward time-shift operator, i. e., $q^{-1} y_k = y_{k-1}$, and I is the identity matrix.

Similar to the SS form, the matrix functions $a_i : \mathcal{P} \to \mathbb{R}^{n_y \times n_y}$, $b_j : \mathcal{P} \to \mathbb{R}^{n_y \times n_u}$, and $c_j : \mathcal{P} \to \mathbb{R}^{n_y \times n_y}$ are assumed to have an affine dependency on θ, e. g., $a(\theta_{k-i}) = \sum_{l=1}^{n_\mu} \mu_{k-i}^{(l)} a_i^{(l)}$ where $a_i^{(l)} \in \mathbb{R}^{n_y \times n_y}$, and $b_0^{(1)} \in \mathbb{R}^{n_y \times n_u}$. Note that the IO polynomials $A(\cdot), B(\cdot), C(\cdot)$ have a specific dynamic dependency on the scheduling parameter vector θ, where the time-shift coincides with the shift of the associated output y, input u, or noise e, respectively. Therefore, this input-output model is in the so called shifted form (SF) [8].

This shifted form has a direct SS realization

$$x_{k+1} = A(\theta_k) x_k + B(\theta_k) u_k + K(\theta_k) e_k, \tag{4a}$$

$$y_k = C(\theta_k) x_k + D(\theta_k) u_k + e_k \;, \tag{4b}$$

where $K(\theta) \in \mathbb{R}^{n_x \times n_y}$ is the observer gain matrix and the matrices $A(\cdot), \ldots, D(\cdot), K(\cdot)$ have the following static, affine dependency structure (in the case of $n_a = n_b = n_c$)

$$A(\theta) = \begin{bmatrix} -a_1(\theta) & I & 0 & \cdots & 0 \\ -a_2(\theta) & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -a_{n_a-1}(\theta) & 0 & \cdots & 0 & I \\ -a_{n_a}(\theta) & 0 & \cdots & \cdots & 0 \end{bmatrix},$$

$$\begin{bmatrix} B(\theta) & K(\theta) \end{bmatrix} = \begin{bmatrix} b_1(\theta) - a_1(\theta) b_0^{(1)} & c_1(\theta) - a_1(\theta) \\ \vdots & \vdots \\ b_{n_a}(\theta) - a_{n_a}(\theta) b_0^{(1)} & c_{n_a}(\theta) - a_{n_a}(\theta) \end{bmatrix},$$

$$C(\theta) = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix}, \qquad D(\theta) = b_0^{(1)}.$$

This state-space model has a static parameter dependency, where due to space limitations, the time index k of the scheduling parameter is omitted. Two other IO forms which have a direct SS realization with static, affine parameter dependency are known, the augmented form (AF) and the observability form (OF) [8], [9]. However, these three simplified SS representations come, most likely, at the cost of a non-minimal state dimension and can only represent a subset of the behaviors associated to the full SS model $G(\theta)$. In this paper, only the SF is presented since it attains the highest accuracy on the identification examples considered in Section V, see [9] for a detailed comparison.

The state order n_x of the resulting SS model (4) depends on multiples of the number of outputs. More specifically, the order is of $n_x = \max(n_a, n_b, n_c) n_y$. For example, for a system with a single input $n_u = 1$ and four outputs $n_y = 4$, the state-space representation acquires order 4, 8 and 12 with increasing values of n_a from 1, 2 and 3, respectively. To find a more economical sized approximation of the SS model, one could compute a set of maximally independent rows of $[A, B]$ [8]; however, these symbolic computations are cumbersome. Therefore, in this paper, we present a fast model order reduction technique for SS models in Section IV.

When examining the SF, it might seem that it can only capture a very restrictive set of behaviors. However, in [8, Sec. IV.D] it was suggested that in the LPV modeling framework there exists a trade-off between allowing more complex dependency structures with smaller state order or simplified dependency structures with increased state order n_x. For example, allowing the parameterization of (3) to

[1]The left inverse is given as $A^\dagger(q, \theta_k) = \sum_{i=0}^{\infty} (I - A(q, \theta_k))^i$, which can be shown based on telescopic sums.

depend rationally on the scheduling signal with low state order n_x versus affine parameterization (3) with higher state order n_x. See [8, Sec. IV.D] for the in-depth treatment and methods to construct these transition matrices, however, a general theorem proving that there always exists such a transformation is not reported in the literature. In this paper, we will not fully develop these symbolic methods, but we provide an example.

Example 1 (Eliminating dynamic dependency): Assume the following IO representation defined LPV system

$$y_k = -\theta_k y_{k-1} + u_k + (1+\theta_k)u_{k-1} \ , \tag{5}$$

has the following state-minimal realization

$$x_{k+1} = -\theta_k x_k + \frac{1}{\theta_{k+1}} u_k \qquad y_k = \theta_k x_k + u_k \ . \tag{6}$$

By introducing an additional state, the following static, affine dependency structure can be found

$$\breve{x}_{k+1} = \begin{bmatrix} -\theta_k & (1+\theta_k) \\ 0 & 0 \end{bmatrix} \breve{x}_k + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u_k, \tag{7a}$$

$$y_k = \begin{bmatrix} -\theta_k & (1+\theta_k) \end{bmatrix} \breve{x}_k + u_k. \tag{7b}$$

Eq. (7) clearly has static affine dependency on the scheduling signal. Hence, dynamic dependency is eliminated by introducing an additional state. Note that there exists no state transformation matrix which can form (6) into a first order SS representation with static, affine dependency. We will verify this claim later on. □

The model (1) is estimated by applying the prediction error method (PEM) framework with an ℓ_2 cost function. This non-linear in the parameter problem is solved by applying the iterative pseudo linear regression [13, Algorithm 3]. First (1) is reformulated in the one-step-ahead predictor

$$\hat{y}_{k|k-1} = (1 - A(q,\theta_k))y_k + B(q,\theta_k)u_k + (1 - C(q,\theta_k))\hat{e}_k, \tag{8}$$

where $\hat{e}_k$ is the estimated noise sequence $\hat{e}_k = y_k - \hat{y}_{k|k-1}$. The algorithm consists of two steps which are performed until convergence: 1) solve the least squares (LS) problem with $\hat{e}_k = y_k - \hat{y}_{k|k-1}$ known from the previous iteration and 2) update $\hat{e}_k$. The highlighted ARMAX model identification method finds a consistent estimate, if the data-generating system coincides with the model set. In this case, a consistent estimate of the process $G(q,\theta)$ and noise $H(q,\theta)$ model can be found. When the noise model is not of interest or no priori information is known, then an output error (OE) model can be used instead. The same consistency result can be shown for OE identification as in the LTI case. Hence, the process model $G(q,\theta)$ is consistently identified, however, it has often a higher parameter variance. Moreover, the noise model $H(q,\theta)$ and, therefore, $K(\theta)$ in (4) are not identified.

B. LPV Subspace Identification

In LPV subspace identification methods, the following innovation form of the to be identified system is considered, e. g., see [10],

$$x_{k+1} = \sum_{l=1}^{n_\mu} \mu_k^{(l)} \left(A^{(l)} x_k + B^{(l)} u_k + K^{(l)} e_k \right), \tag{9a}$$

$$y_k = C x_k + D u_k + e_k \ . \tag{9b}$$

For SID, the number of past time increments p (past window) and future time increments f have to be chosen a-priori[2] under the condition $n_x \leq f n_y \leq p n_y$. The past and future window indirectly provide the order of the identified IO model from which an SS model is realized. In this paper, the LPV PBSID$_{\text{opt}}$ subspace identification algorithm from the PBSID toolbox is used [10].

C. Comparing both Identification Methods

Subspace identification with the innovation form (9) assumes a white noise innovation error e_k on the states and output signals, respectively. Finding the convolution the state equation in terms of u, y or u, e leads to an infinite-order IO autoregressive with exogenous input (ARX) or moving average with exogenous input (MAX) model, depending if the state-space model is considered in innovation form as in (9) or in predictor form [14], [15]. So, the IO model in (1) is not equal to the innovation form and they have different representation capabilities. However, the ARX or MAX models in SID come with an exponential growing parameterization. To reduce the complexity, we have a specific dynamic dependency structure with a significant smaller amount of parameters of the IO polynomials $A(\cdot), B(\cdot), C(\cdot)$ in (3) compared to the high order ARX or MAX parameterizations in SID at the cost of limited representation capabilities. Hence, we assume that the dominant dynamics of the underlying system can be captured with this simplified ARMAX model. If one would increase the model complexity of the ARMAX model to include different dynamic dependency structures, it can coincide with the ARX and MAX models in SID at a certain point. Yet, one should realize that the IO model in (1) has different representation capabilities compared to the fully parameterized innovation form. How the simplified ARMAX structure (1) and the ARX or MAX model in SID exactly concur is for future research. Alternatively, we might get even a lower order IO model, if an LPV Box-Jenkins (BJ) representation is used. However, the direct realization of the BJ to SS results, in general, in unwanted dynamic and rational dependencies in the SS form [7].

In the MIMO case, the SS model representations have parsimonious parameterization compared to IO models, as shared dynamics among the outputs can be represented by a lower order subspace. As a tool to obtain an appropriate state dimension, in subspace identification, the model order n_x can be chosen by the user or it can be specified automatically, which will be achieved by a singular value decomposition (SVD) and detecting a gap in the singular values (see [12, Step 4 of Algorithms 1 and 2]). This can also be viewed as a model order reduction scheme directly performed on the identified IO model. However, we will show that in the LPV case, due to the complexity of subspace estimation, it can still be favorable from a computational point of view to estimate the simplified IO structure (2) with direct SS realization and apply model order reduction to get an accurate, low order estimate.

[2]Best identification results in the examples in Section V are obtained if f is chosen as p or $p-1$.

IV. MODEL ORDER REDUCTION

In this section, a model order reduction scheme is presented to find the state-minimal realization of (1) and is able to apply model order truncation.

In the literature, many model order reduction schemes are known, e. g., see [16]. From the LTI approximate approaches, the coprime factor [17], optimal Hankel norm and balanced truncation [18], [19], and Ho-Kalman like realization [8] methods are extended to the LPV case. In this paper, we are focusing on system identification where identification of unstable systems is common. Model order reduction of unstable systems via standard balanced realization methods is impossible, as quadratic stabilizability and detectability of the system is required. Coprime factorization and the Ho-Kalman like realization are capable to deal with both stable and unstable models, however, both methods are accompanied by a significant computational load in the LPV case [8], [17]. Therefore, in the following, we will present a low complexity scheme based on the ideas of [8] to obtain the state-minimal realization or to apply model order reduction.

To start, as in [8], based on the state-space matrices given in (4), define the matrices

$$\mathcal{M}_1 = \begin{bmatrix} B^{(1)} & \dots & B^{(n_\mu)} & K^{(1)} & \dots & K^{(n_\mu)} \end{bmatrix}, \quad (10a)$$

$$\mathcal{M}_k = \begin{bmatrix} A^{(1)}\mathcal{M}_{k-1} & \dots & A^{(n_\mu)}\mathcal{M}_{k-1} \end{bmatrix}, \quad (10b)$$

and the j-step extended reachability matrix

$$\mathcal{R}_j = \begin{bmatrix} \mathcal{M}_1 & \dots & \mathcal{M}_j \end{bmatrix} \in \mathbb{R}^{n_x \times \left((n_u+n_y)\sum_{k=1}^{j} n_\mu^k\right)}. \quad (11)$$

Similarly, define

$$\mathcal{N}_1 = C^{(1)}, \quad (12a)$$

$$\mathcal{N}_k^\top = \begin{bmatrix} (\mathcal{N}_{k-1}A^{(1)})^\top & \dots & (\mathcal{N}_{k-1}A^{(n_\mu)})^\top \end{bmatrix}, \quad (12b)$$

to build the i-step extended observability matrix

$$\mathcal{O}_i = \begin{bmatrix} \mathcal{N}_1^\top & \dots & \mathcal{N}_i^\top \end{bmatrix}^\top \in \mathbb{R}^{\left(n_y(1+\sum_{k=2}^{i} n_\mu^k)\right)\times n_x}. \quad (13)$$

Consequently, the extended Hankel matrix is given by

$$\mathcal{H}_{ij} = \mathcal{O}_i\mathcal{R}_j \in \mathbb{R}^{\left(n_y(1+\sum_{k=2}^{i} n_\mu^k)\right)\times\left((n_u+n_y)\sum_{k=1}^{j} n_\mu^k\right)}. \quad (14)$$

A realization of the SS can be found by applying the SVD on the extended Hankel matrix (14), which reveals underlying model order. A state-minimal realization can be obtained by selecting all non-zero singular values [20], which amount is n_G, i. e., $\mathrm{rank}(\mathcal{H}_{n_x n_x}) = n_\mathrm{G}$ [20, Theorem 2] (if the system is state-minimal, then $n_\mathrm{G} = n_x$). A lower order approximation of the SS representation (4) can be found by selecting only the first $n_{\hat{x}} \leq n_\mathrm{G}$ singular values [8]. The LPV-SS representation $G(\theta)$ is called state-minimal, if there exists no other LPV-SS representation $G'(\theta)$ with $n'_x < n_\mathrm{G}$ and has equivalent IO behavior (see [20, Theorem 1] for details). Hence, we define state-minimality w. r. t. the class of SS models with static, affine dependency structure.

In subspace identification, the (extended) Hankel matrix is the key ingredient to find a realization of the state or the state-space matrices [10], [21]. As a downside of using (14) directly, the size of the extended Hankel matrix increases exponentially with i and j leading easily to numerical limitations of the Ho-Kalman technique. In case the extended Hankel matrix is constructed from an estimated model, i and j need to be chosen large enough such that $\mathrm{rank}(\mathcal{H}_{ij}) \geq n_\mathrm{G}$, where it is often preferable to choose higher i and j for a more accurate reduced model [22].

To overcome this numerical problem, a bases reduced algorithm was proposed recently [21]. The idea is to select only the non repetitive parts of the extended Hankel matrix, which will reduce the computational load. The reduced basis results in smaller sized sub-Hankel matrices on which matrix decompositions are performed. Unfortunately, due to the structure of $\mathcal{O}_i$ and $\mathcal{R}_j$ for (4) there are no obvious bases that can always be avoided. Hence, there should be an automated method of selecting bases until certain condition holds. However, how to do this selection in a computationally efficient and automated manner is still an open question.

Therefore, in this paper, we will decrease the computational load to obtain the SVD of the full Hankel matrix significantly by applying QR decompositions on the observability and reachability matrix. More specifically, compute the following two QR decompositions $\mathcal{O}_i = Q_\mathcal{O}R_\mathcal{O}$ and $\mathcal{R}_j^\top = Q_\mathcal{R}R_\mathcal{R}$. Consequently, a SVD of only $R_\mathcal{O}R_\mathcal{R}^\top \in \mathbb{R}^{n_x\times n_x}$ is required

$$\mathcal{H}_{ij} = \mathcal{O}_i\mathcal{R}_j = Q_\mathcal{O}R_\mathcal{O}R_\mathcal{R}^\top Q_\mathcal{R}^\top = Q_\mathcal{O}U\Sigma V^\top Q_\mathcal{R}^\top. \quad (15)$$

Opposed to the LPV Ho-Kalman like scheme, we will find then a realization by finding the static projection matrix and also avoid the computation of the shifted Hankel matrix.

Lemma 1 (Fast Ho-Kalman reduction scheme): Choose i and j large enough such that $\mathrm{rank}(\mathcal{O}_i) = n_\mathrm{G}$ and $\mathrm{rank}(\mathcal{R}_j) = n_\mathrm{G}$, respectively. Perform the following decompositions $\mathcal{O}_i = Q_\mathcal{O}R_\mathcal{O}$, $\mathcal{R}_j^\top = Q_\mathcal{R}R_\mathcal{R}$, and $R_\mathcal{O}R_\mathcal{R}^\top = U\Sigma V^\top$. Then a realization of the state-space matrices is found by

$$\hat{A}^{(l)} = T^\dagger A^{(l)} T, \qquad \hat{B}^{(l)} = T^\dagger B^{(l)}, \quad (16)$$

$$\hat{C}^{(1)} = C^{(1)} T, \qquad \hat{K}^{(l)} = T^\dagger K^{(l)}, \quad (17)$$

where $l = 1, \dots, n_\mu$ and the projection matrix is

$$T = R_\mathcal{R}^\top V_{n_{\hat{x}}}\Sigma_{n_{\hat{x}}}^{-\frac{1}{2}}, \qquad T^\dagger = \Sigma_{n_{\hat{x}}}^{-\frac{1}{2}} U_{n_{\hat{x}}}^\top R_\mathcal{O}, \quad (18)$$

where U_n, V_n denotes the first n columns of the matrices U, V, respectively, Σ_n denotes the upper n by n matrix of Σ, and $1 \leq n_{\hat{x}} \leq n_\mathrm{G}$. In (18), $T^\dagger$ is the left pseudo inverse of the projection matrix T, i. e., $T^\dagger T = I_{n_{\hat{x}}}$. □

Proof: In the Ho-Kalman like scheme, the shifted matrix is chosen as:

$$\overleftarrow{\mathcal{H}}_{ij} = \mathcal{O}_i \begin{bmatrix} A^{(1)} & \dots & A^{(n_\mu)} \end{bmatrix} (I_{n_\mu} \circledcirc \mathcal{R}_j), \quad (19)$$

where the block-wise Kronecker product $\circledcirc$ is defined as

$$(I_{n_\mu} \circledcirc \mathcal{R}_j) = \begin{bmatrix} I_{n_\mu}\otimes\mathcal{M}_1 & \dots & I_{n_\mu}\otimes\mathcal{M}_j \end{bmatrix}. \quad (20)$$

In our scheme, the observability and reachability matrices are taken $\hat{\mathcal{O}}_i = Q_\mathcal{O}U_{n_{\hat{x}}}\Sigma_{n_{\hat{x}}}^{\frac{1}{2}}$ and $\hat{\mathcal{R}}_j = \Sigma_{n_{\hat{x}}}^{\frac{1}{2}}V_{n_{\hat{x}}}^\top Q_\mathcal{R}^\top$. Then,

$$\begin{aligned} &\hat{\mathcal{O}}_i^\dagger \overleftarrow{\mathcal{H}}_{ij}(I_{n_\mu} \circledcirc \hat{\mathcal{R}}_j^\dagger) \\ &\quad = \Sigma_{n_{\hat{x}}}^{-\frac{1}{2}} U_{n_{\hat{x}}}^\top Q_\mathcal{O}^\top Q_\mathcal{O} R_\mathcal{O} \begin{bmatrix} A^{(1)} & \dots & A^{(n_\mu)} \end{bmatrix} \cdot \\ &\qquad (I_{n_\mu} \circledcirc R_\mathcal{R}^\top Q_\mathcal{R}^\top Q_\mathcal{R} V_{n_{\hat{x}}}\Sigma_{n_{\hat{x}}}^{-\frac{1}{2}}) \\ &\quad = T^\dagger \begin{bmatrix} A^{(1)} & \dots & A^{(n_\mu)} \end{bmatrix} (I_{n_\mu} \circledcirc T). \end{aligned} \quad (21)$$

Hence, the construction of the $\hat{A}^{(l)}$ matrices is proven. Recall that $C^{(1)}$ is constructed from the first n_y columns of the Hankel matrix, i. e., $\mathcal{H}_{1j} = C^{(1)}\mathcal{R}_j$. Therefore, $\hat{C}^{(1)}$ is

$$\mathcal{H}_{1j}\hat{\mathcal{R}}_j^{\dagger} = C^{(1)}\mathcal{R}_j\hat{\mathcal{R}}_j^{\dagger} = C^{(1)}T = \hat{C}^{(1)} \ . \tag{22}$$

Hence, the construction of $\hat{C}^{(1)}$ is provided. Similar argument can be made for the construction of $\hat{B}^{(l)}$, $\hat{K}^{(l)}$ and is therefore omitted. Finalizing, we show that $T^{\dagger}T = I_{n_{\hat{x}}}$. Note that $U_{n_{\hat{x}}}^{\top}U = \begin{bmatrix} I_{n_{\hat{x}}} & 0 \end{bmatrix}$, $V_{n_{\hat{x}}}^{\top}V = \begin{bmatrix} I_{n_{\hat{x}}} & 0 \end{bmatrix}$ for any $n_{\hat{x}}$ (orthogonality property). Hence, $T^{\dagger}T = \Sigma_{n_{\hat{x}}}^{-\frac{1}{2}}U_{n_{\hat{x}}}^{\top}R_{\mathcal{O}}R_{\mathcal{R}}^{\top}V_{n_{\hat{x}}}\Sigma_{n_{\hat{x}}}^{-\frac{1}{2}} = \Sigma_{n_{\hat{x}}}^{-\frac{1}{2}}\begin{bmatrix} I_{n_{\hat{x}}} & 0 \end{bmatrix}\Sigma\begin{bmatrix} I_{n_{\hat{x}}} & 0 \end{bmatrix}^{\top}\Sigma_{n_{\hat{x}}}^{-\frac{1}{2}} = I_{n_{\hat{x}}}$. ■

Note that in the original LPV Ho-Kalman like method the shifted Hankel matrix (19) is constructed from $\mathcal{R}_{j-1}$ [8, Eq. (51a)] and we apply $\mathcal{R}_j$. However, this modification does not change the realization in any way. Besides, note that $C^{(l)}$ for $2 \leq l \leq n_{\mu}$ is absent, because the shifted SS form (4) has a parameter independent C matrix function. However, for general SS model reduction, adding $C^{(l)}$ is straightforward by extending $\mathcal{N}_1$ (12a) with $C^{(l)}$ and then $\hat{C}^{(l)} = C^{(l)}T$ in Lemma 1.

Additionally, note that Lemma 1 only requires the $R_{\mathcal{O}}$ and $R_{\mathcal{R}}$ matrices, which are efficiently computed by a UDU decomposition of $\mathcal{O}_i^{\top}\mathcal{O}_i = U_{\mathcal{O}}^{\top}D_{\mathcal{O}}U_{\mathcal{O}} \in \mathbb{R}^{n_x \times n_x}$ and $\mathcal{R}_j\mathcal{R}_j^{\top} = U_{\mathcal{R}}^{\top}D_{\mathcal{R}}U_{\mathcal{R}} \in \mathbb{R}^{n_x \times n_x}$ where $R_{\mathcal{O}} = D_{\mathcal{O}}^{\frac{1}{2}}U_{\mathcal{O}}$ and $R_{\mathcal{R}} = D_{\mathcal{R}}^{\frac{1}{2}}U_{\mathcal{R}}$, respectively. The pre-decomposition of the observability and reachability matrix is useful from two view-points. Firstly, it avoids construction of the exponentially growing Hankel matrix and shifted Hankel matrix. Secondly, the UDU factorization is computationally cheaper then the SVD [23] and Lemma 1 performs the SVD on a much smaller sized matrix, i. e., performs the SVD on $R_{\mathcal{O}}R_{\mathcal{R}}^{\top} \in \mathbb{R}^{n_x \times n_x}$ and not $\mathcal{H}_{ij} \in \mathbb{R}^{\left(n_y(1+\sum_{k=2}^{i} n_{\mu}^k)\right)\times\left((n_u+n_y)\sum_{k=1}^{j} n_{\mu}^k\right)}$.

Theoretical global error bounds between the original and model order reduced representation have not been developed yet (as known in the LTI case [16]) and remain objectives for future research. However, approximate error bounds can be computed using the μ-test in the Enhanced linear fractional transformation toolbox [24].

The highlighted scheme is a modified version of the Ho-Kalman scheme in [8], which is proven to provide a state-minimal representation [20, Corollary 1] if $n_{\hat{x}} = n_{\mathrm{G}}$. Hence, lets verify our claim of state-minimality of the representation (7) in Example 1.

Example 1 (cont'd): Lets verify if (7) is a state-minimal static, affine representation. See that

$$A^{(1)} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \ A^{(2)} = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}, \ B^{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$
$$C^{(1)} = \begin{bmatrix} 0 & 1 \end{bmatrix}, \ C^{(2)} = \begin{bmatrix} -1 & 1 \end{bmatrix},$$

which provides the following structural matrices

$$\mathcal{R}_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad \mathrm{rank}(\mathcal{R}_1) = 2,$$

$$\mathcal{O}_1 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & -1 & 0 & -1 \end{bmatrix}^{\top}, \quad \mathrm{rank}(\mathcal{O}_1) = 2,$$

$$\mathcal{H}_{11} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\top}, \quad \mathrm{rank}(\mathcal{H}_{11}) = 2.$$

Hence, direct application of [20, Theorem 1] and $\mathrm{rank}(\mathcal{H}_{11}) = 2$ it is concluded that (7) is a state-minimal static affine representation. □

Next, we will highlight an example where Lemma 1 is applied to find a state-minimal realization.

Example 2 (Shared dynamics): The following shifted IO form is identified

$$y_k = \begin{bmatrix} 1+\theta_{k-1} & 0 \\ 0 & 1 \end{bmatrix} y_{k-1} + \begin{bmatrix} \theta_{k-2} & \theta_{k-2} \\ -\theta_{k-2} & -\theta_{k-2} \end{bmatrix} y_{k-2} + \begin{bmatrix} 1+\theta_{k-1} & 1 \\ 1 & 1+\theta_{k-1} \end{bmatrix} u_{k-1}. \tag{23}$$

Using the direct realization (4), the state-space form has a state dimension of 4. The extended observability matrix and reachability matrix will not be displayed because of space limitations, however, it is straightforward to compute $\mathrm{rank}(\mathcal{R}_4) = 3$, $\mathrm{rank}(\mathcal{O}_4) = 4$, and $\mathrm{rank}(\mathcal{H}_{44}) = 3$. Hence, the realized system is not span-reachable of order 4 ([20, Theorem 2]). Therefore, the minimal state dimension for the static, affine representation is $n_x = 3$ [20, Corollary 1] and a state-minimal representation can be found by applying the (fast) Ho-Kalman realization. The provided model order reduction scheme returns the (numerically rounded) system matrices:

$$A^{(1)} = \begin{bmatrix} 1.17 & 0.15 & -0.37 \\ -0.06 & 0.94 & 0.13 \\ 0.53 & 0.47 & -0.11 \end{bmatrix}, \quad B^{(1)} = \begin{bmatrix} 0.64 & 0.06 \\ -0.33 & -0.68 \\ 0.16 & -0.26 \end{bmatrix},$$
$$A^{(2)} = \begin{bmatrix} 1.14 & -0.49 & 0.36 \\ 0.05 & -0.46 & -0.48 \\ -0.21 & 0.43 & 0.32 \end{bmatrix}, \ B^{(2)} = \begin{bmatrix} 0.70 & 0.70 \\ -1.01 & -1.01 \\ -0.09 & -0.09 \end{bmatrix},$$
$$C^{(1)} = \begin{bmatrix} 1.26 & -0.19 & 0.80 \\ -0.19 & -1.00 & -1.29 \end{bmatrix}.$$

□

Computational complexity

The amount of unknown parameters is

$$N_{\text{unknowns}} = n_y n_y n_{\mu}(n_a + n_c) + n_y n_u(1 + n_{\mu} n_b)$$

in the SF model (3). The parameterization is clearly linear in all design variables, except in the output dimension n_y, which it is quadratic. The model is identified with an iterative LS method. To the authors experience, only few iterations are need to be taken before convergence. To compute the approximate model order, two UDU factorizations are performed on $\mathcal{O}_i^{\top}\mathcal{O}_i \in \mathbb{R}^{n_x \times n_x}$ and $\mathcal{R}_j\mathcal{R}_j^{\top} \in \mathbb{R}^{n_x \times n_x}$ where a UDU factorization needs $n_x^3/3$ flops [23, Algorithm 4.2.2]. Subsequently, the SVD is performed on $R_{\mathcal{O}}R_{\mathcal{R}}^{\top} \in \mathbb{R}^{n_x \times n_x}$ where a SVD needs $21n_x^3$ [23, Fig. 8.6.1]. To compare, the PBSID$_{\text{opt}}$ method [10] has

$$N_{\text{unknowns}} = n_y \left(n_u + (n_y + n_u)\sum_{j=1}^{p} n_{\mu}^j \right) ,$$

which is linear in n_u, quadratic in n_y, and exponential in n_μ (with p such that $n_y p \geq n_x$)[3]. Advantageously, the $\text{PBSID}_{\text{opt}}$ only needs one LS step. To estimate the state sequence, a SVD is performed on the matrix $\Gamma^p \mathcal{K}^p Z \in \mathbb{R}^{n_y p \times N}$ [10, Eq. (15)] with $4N^2 n_y p + 8N(n_y p)^2 + 9(n_y p)^3$ flops[4]. Hence, we can conclude that our proposed estimation scheme scales significantly better w. r. t. the design parameters.

V. EXAMPLES

In this section a numerical and an experimental example are presented. The best fit rate (BFR), see [7, Definition 2.4], of the identified models using validation data is computed.

A. Numerical Example

A fourth order LPV MIMO plant in the form of (9) introduced in [11] is tested with two inputs, three outputs and three scheduling parameters. For the estimation and validation data, the input, scheduling parameter, and noise signals are designed as in [11], with an averaged signal-to-noise ratio of about 20 dB. The validation outputs are simulated without noise, in order to evaluate the simulated output of the identified model. The identification results are shown in Fig. 1 in terms of BFR versus model order.

To find the best LPV model for each order, in SID, both parameter dependent and independent input and observer gain matrices are considered, i. e., $B(\mu) = B$ and $K(\mu) = K$, respectively. Also, for each past window $p \leq 6$ different models of order up to pn_y are identified. Further settings (e. g., regularization) are chosen as suggested in the PBSID toolbox documentation. Then, in Fig. 1 only the best results are plotted, i. e., the highest mean fit for a given order. In the IO identification case, first an ARX model is fitted ($n_c = 0$) as in [9] with all combinations of n_a and n_b with $n_b \leq n_a \leq 6$. Since the model order depends only on $n_a n_y$, the same model order is obtained for different $n_b \leq n_a$ and again the best results are plotted in Fig. 1. Furthermore, ARMAX models are identified using the IO method with PEM, see Section III. Now, also the order n_c is varied again with the limitation $n_c \leq n_a$. Hence, we identify various IO models with the iterative algorithm with equivalent model order and we only display the model with the highest BFR. Fig. 1 shows that this method outperforms the (non-iterative) SID for high model orders.

Using IO identification, it is not possible to identify a model of the same order as that of the original plant without performing model reduction. Applying Lemma 1 to each one of the identified ARMAX models results in LPV models of any order up to 18. Then by choosing the best BFR for a given model order over all ARMAX and order reduced models, the purple line denoted by ARMAX $\mathcal{H}_{ij}$ is obtained. Hence, it is clear that our proposed ARMAX method has similar performance to $\text{PBSID}_{\text{opt}}$ on this particular example while keeping computational complexity low.

[3]Using the Kernel method, N_{unknowns} can be limited to $n_y N$, however, regularization is required for this ill-conditioned matrix. Hence, additional steps are necessary to obtain an appropriate regularization parameter [10, Sec. 5.3].

[4]With the estimated state sequence, an additional LS step is executed to estimate the system matrices. We will disregard this step in our analysis.

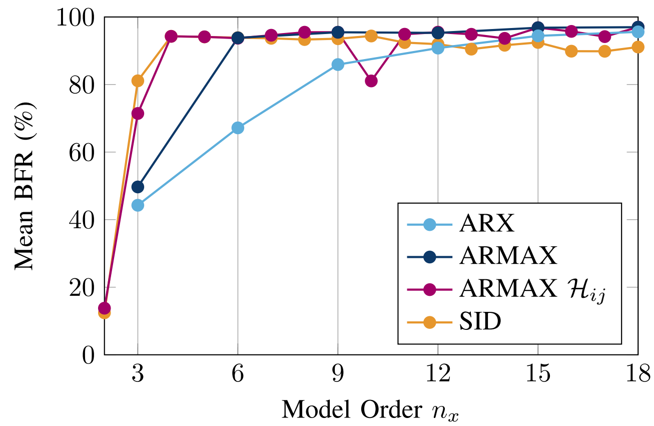

Fig. 1. Identification results on fourth order MIMO example with averaged fit using IO identification with ARX and ARMAX model structures as well as SID, where the model order reduction scheme is applied to the identified ARMAX models

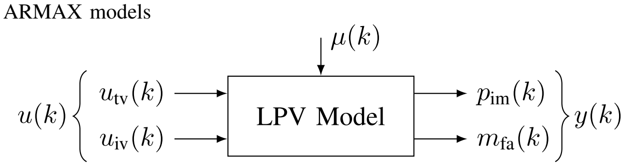

Fig. 2. Air-Path system identified in this Section with unknown scheduling parameter vector

B. Air-Path

In internal combustion engines the number of actuators, e. g., throttle, intake, and outlet valves, increases to improve the efficiency and the dynamic torque generation. The generated torque depends directly on the amount of fresh air m_{fa} in the cylinder, where the air-path denotes the flow path that the air is taking from the environment into the cylinder and which can be influenced by the throttle and inlet valve. In order to control this system, the idea is to synthesize an LPV controller, thus an LPV model of the air-path is required. In Fig. 2, the setup for the identification is shown schematically, which is similar to this considered in [9]. The actuators (inputs) to control the fresh air entering the cylinder are the throttle valve position u_{tv} and the intake valve position u_{iv}. The controlled outputs are the amount of fresh air inside the cylinder m_{fa} and the intake manifold pressure p_{im}. Measurements were obtained from a test rig at IAV GmbH. In contrast to [9], the measurements are taken with a time-varying engine speed n_{eng}. These time variations make identification more complex to the setting in [9]. The measurements were sampled at 0.01 s and different candidate parameters for the scheduling parameters were chosen and assessed, since it is unknown which scheduling signal should be used. The following scheduling signal is used $\mu^\top = \begin{bmatrix} 1 & n_{\text{eng}} & u_{\text{tv}} & p_{\text{im}} \end{bmatrix}$, obtaining the most accurate models, and the identification results are shown in Fig. 3.

For model orders up to three, SID gives better results compared to the IO identification using ARX and ARMAX model structures. However, the accuracy of the models obtained by LPV-IO identification, in contrast to SID, is increasing with the model order, i. e., past window p or

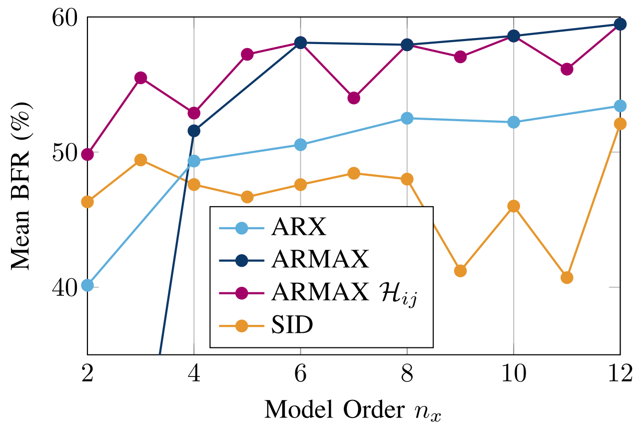

Fig. 3. Identification results on the air-path system of a gasoline engine with averaged fit using LPV-IO methods (ARX and ARMAX model structures) and also the LPV-SID method, where the model order reduction scheme is applied to the identified ARMAX models

filter order n_a, respectively. Furthermore, reducing the model order of the ARMAX models leads even to superior BFRs for low orders compared to SID. Besides, the BFR of the identified LPV SID models is inferior to the results shown in [9]. However, this originates from engine speed n_{eng}, which is here time-varying while it is kept constant in [9].

VI. CONCLUSION

In this paper, a procedure is presented to identify LPV-SS models with static and affine dependency on the scheduling signal in a computational efficient manner. To this end, an LPV ARMAX model is identified with a specific dynamic dependency on the scheduling parameters, which has a direct LPV-SS representation with static, affine parameter dependency. The model complexity is reduced by allowing a different parameterization with respect to LPV-SID algorithms. However, in the MIMO case, this LPV-SS model will most likely result in a non-minimal state representation. Therefore, a novel, fast model order reduction scheme is suggested based on the Ho-Kalman algorithm to obtain a low order model. It is shown that the computational complexity of this Ho-Kalman like scheme is reduced tremendously compared to the original scheme. Comparable steps are conducted in state-of-the-art LPV-SID methods, but we showed that it is computationally more intensive and our method scales better w. r. t. number of samples, the input, scheduling, and output dimensions. Two examples are included to compare the proposed method with the $\text{PBSID}_{\text{opt}}$ algorithm. In the first numerical toy example, it has been shown that the proposed IO scheme and $\text{PBSID}_{\text{opt}}$ are comparable in performance. The second example was the estimation of a real-life model of the air-path of a gasoline engine, where the proposed method clearly outperforms SID. Hence, the proposed two step scheme is a step forward in making LPV-SS identification schemes practically applicable, due to its low computational complexity.

REFERENCES

[1] C. Hoffmann and H. Werner, “A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations,” *IEEE Trans. on Control Systems Technology*, vol. 23, no. 2, pp. 416–433, 2015.

[2] I. Masubuchi, J. Kato, M. Saeki, and A. Ohara, “Gain-scheduled controller design based on descriptor representation of LPV systems: application to flight vehicle control,” in *Proc. of the 43rd Conf. on Decision and Control*, Atlantis, Bahamas, Dec. 2004, pp. 815–820.

[3] R. Zope, J. Mohammadpour, K. Grigoriadis, and M. Franchek, “Robust fueling strategy for an SI engine modeled as an linear parameter varying time-delayed system,” in *Proc. of the American Control Conf.*, Baltimore, MD, USA, Jun. 2010, pp. 4634–4639.

[4] Z. Yu, H. Chen, and P.-y. Woo, “Gain scheduled LPV $\mathcal{H}_\infty$ control based on LMI approach for a robotic manipulator,” *J. of Robotic Systems*, vol. 19, no. 12, pp. 585–593, 2002.

[5] B. Bamieh and L. Giarre, “Identification of linear parameter varying models,” *Int. J. of Robust and Nonlinear Control*, vol. 12, no. 9, pp. 841–853, 2002.

[6] M. Butcher, A. Karimi, and R. Longchamp, “On the consistency of certain identification methods for linear parameter varying systems,” in *Proc. of the 17th IFAC World Congress*, Seoul, Korea, Jul. 2008, pp. 4018–4023.

[7] R. Tóth, *Modeling and identification of linear parameter-varying systems*. Springer, 2010, vol. 403.

[8] R. Tóth, H. S. Abbas, and H. Werner, “On the state-space realization of LPV input-output models: Practical approaches,” *IEEE Trans. on Control Systems Technology*, vol. 20, no. 1, pp. 139–153, 2012.

[9] E. Schulz, A. Bussa, and H. Werner, “Identification of linear parameterivarying systems via IO and subspace identification - a comparison,” in *Proc. of the 55th Conf. on Decision and Control*, Las Vegas, NV, USA, Dec. 2016, pp. 7147–7152.

[10] J.-W. van Wingerden and M. Verhaegen, “Subspace identification of bilinear and LPV systems for open-and closed-loop data,” *Automatica*, vol. 45, no. 2, pp. 372–381, 2009.

[11] V. Verdult and M. Verhaegen, “Subspace identification of multivariable linear parameter-varying systems,” *Automatica*, vol. 38, no. 5, pp. 805–814, 2002.

[12] ——, “Kernel methods for subspace identification of multivariable LPV and bilinear systems,” *Automatica*, vol. 41, no. 9, pp. 1557–1565, 2005.

[13] R. Tóth, P. S. C. Heuberger, and P. M. J. Van den Hof, “Prediction-error identification of LPV systems: present and beyond,” in *Control of Linear Parameter Varying Systems with Applications*, J. Mohammadpour and C. W. Scherer, Eds. Springer, 2012, ch. 2, pp. 27–58.

[14] P. B. Cox and R. Tóth, “Alternative form of predictor based identification of LPV-SS models with innovation noise,” in *Proc. of the 55th Conf. on Decision and Control*, Las Vegas, NV, USA, Dec. 2016.

[15] S. J. Qin, “An overview of subspace identification,” *Computers & Chemical Engineering*, vol. 30, no. 10, pp. 1502–1513, 2006.

[16] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. Society for Industrial and Applied Mathematics, 2008.

[17] C. Beck, “Coprime factors reduction methods for linear parameter varying and uncertain systems,” *Systems and Control Letters*, vol. 55, pp. 199–213, 2006.

[18] G. D. Wood, “Control of Parameter-Dependent Mechanical Systems,” PhD Thesis, University of Cambridge, 1995.

[19] G. D. Wood, P. J. Goddard, and K. Glover, “Approximation of linear parameter-varying systems,” in *Proc. of the 35th Conf. on Decision and Control*, Kobe, Japan, dec. 1996, pp. 406–411.

[20] M. Petreczky, R. Tóth, and G. Mercére, “Realization theory for LPV state-space representations with affine dependence,” *IEEE Trans. on Automatic Control*, 2017.

[21] P. B. Cox, R. Tóth, and M. Petreczky, “Estimation of LPV-SS models with static dependency using correlation analysis,” in *Proc. of the 1st IFAC Workshop on Linear Parameter Varying Systems*, Grenoble, France, Oct. 2015, pp. 91–96.

[22] S. Y. Kung, “A new identification and model reduction algorithm via singular value decompositions,” in *Proc. of the 12th Asilomar Conf. on Circuits, Systems and Computers*, Pacific Grove, CA, USA, Nov. 1978, pp. 705–714.

[23] G. Golub and C. Van Loan, *Matrix Computations*, 4th ed. The Johns Hopkins University Press, 2013.

[24] J. F. Magni, “Linear fractional representation toolbox for use with matlab,” available with the SMAC Toolbox at http://w3.onera.fr/smac/lfrt., Feb. 2006.