# On the State-Space Realization of LPV Input-Output Models: Practical Approaches

R. Tóth, *Member, IEEE*, H. S. Abbas, H. Werner, *Member, IEEE*

**Abstract**

A common problem in the context of Linear Parameter-Varying (LPV) systems is how Input-Output (IO) models can be efficiently realized in terms of State-Space (SS) representations. The problem originates from the fact that in the LPV literature discrete-time identification and modeling of LPV systems is often accomplished via IO model structures. However, to utilize these LPV-IO models for control synthesis, commonly it is required to transform them into an equivalent SS form. In general, such a transformation is complicated due to the phenomenon of dynamic dependence (dependence of the resulting representation on time-shifted versions of the scheduling signal). This conversion problem is revisited and practically applicable approaches are suggested which result in discrete-time SS representations that have only static dependence (dependence on the instantaneous value of the scheduling signal). To circumvent complexity, a criterion is also established to decide when an LTI type of realization approach can be used without introducing significant approximation error. To reduce the order of the resulting SS realization, a LPV Ho-Kalman type of model reduction approach is introduced, which, besides its simplicity, is capable of reducing even non-stable plants. The proposed approaches are illustrated by application oriented examples.

**Index Terms**

Linear parameter-varying systems; Realization; Model reduction; Input-output representation; State-space representation; Dynamic dependence.

## I. Introduction

The framework of *Linear Parameter-Varying* (LPV) systems provides an efficient alternative for modeling and control of nonlinear/time-varying systems, proven by a wide range of successful

R. Tóth is with the Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands, email: r.toth@tudelft.nl, H. S. Abbas is with the Electrical Engineering Department, Faculty of Engineering, Assiut University, 71515 Assiut, Egypt, email: h_abbas@aun.edu.eg and H. Werner is with the Institute of Control Systems, Hamburg University of Technology, Eissendorferstr. 40, 21073, Hamburg, Germany, email: h.werner@tuhh.de.

applications from aircrafts [1] to environmental modeling [2]. The practical use of the LPV framework comes from the fact that it offers powerful controller synthesis tools based on the extension of *Linear Time-Invariant* (LTI) approaches, see e.g. [3]–[6].

In the LPV control literature, a discrete-time LPV system is commonly described in a *State-Space* (SS) representation:

$$qx = A(p)x + B(p)u, \tag{1a}$$

$$y = C(p)x + D(p)u, \tag{1b}$$

where $u : \mathbb{Z} \to \mathbb{R}^{n_\mathrm{u}}$, $y : \mathbb{Z} \to \mathbb{R}^{n_\mathrm{y}}$ and $x : \mathbb{Z} \to \mathbb{R}^{n_\mathrm{x}}$ are the input, output and state signals of the system respectively, $q$ is the forward time-shift operator, i.e. $qx(k) = x(k+1)$, and the system matrices $A, B, C, D$ are rational functions of the scheduling signal $p : \mathbb{Z} \to \mathbb{P}$ and nonsingular on $\mathbb{P}$, where the set $\mathbb{P} \subseteq \mathbb{R}^{n_\mathrm{p}}$ is the so called *scheduling space*. It is assumed that $p$ is an external signal of the system and it is online measurable during operation. In case, $p$ is a function of the inputs, outputs or states of (1a-b), then the LPV system is referred to as a *quasi*-LPV system. Note, that all matrices in (1a-b), defined as

$$\left[\begin{array}{c|c} A(p) & B(p) \\ \hline C(p) & D(p) \end{array}\right] : \mathbb{P} \to \left[\begin{array}{c|c} \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}} & \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{u}} \\ \hline \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{x}} & \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{u}} \end{array}\right], \tag{2}$$

are dependent on the instantaneous value of $p$, which is called *static* dependence.

In the LPV literature, numerous approaches have been introduced to identify or to model LPV systems based on various model structures and representations, see [7] for a recent overview and comparison of these methods. A large class of the available approaches, like [8]–[11], addresses the identification problem of LPV systems in terms of so called *Input-Output* (IO) model structures with many applied results like [2], [12]–[14]. In this class of LPV-IO identification methods, the deterministic part of the data-generating system is commonly described in an LPV-IO filter form:

$$y = -\sum_{i=1}^{n_\mathrm{a}} a_i(p) q^{-i} y + \sum_{j=0}^{n_\mathrm{b}} b_j(p) q^{-j} u, \tag{3}$$

where $a_i : \mathbb{P} \to \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{y}}$ and $b_j : \mathbb{P} \to \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{u}}$ are rational matrix functions of $p$ with no singularity on $\mathbb{P}$ and $n_\mathrm{a} \geq n_\mathrm{b} \geq 0$. However, the main stream of LPV control synthesis approaches is derived for SS representations, thus to utilize obtained LPV-IO models in the form of (3) for control synthesis, commonly it is required to transform (3) to an equivalent SS form (1a-b). Additionally, in terms of LPV modeling based on first-principle nonlinear differential equations it is also mathematically attractive to first find a LPV realization of the underlaying behavior in terms of an IO representation and then find a SS realization of the IO form, see [7], [15].

It has been recently observed that representations (1a-b) and (3) are not equivalent in terms of IO-behavior, i.e. in general, an LPV-IO representation (3) cannot be transformed into (1a-b) without deforming the dynamical relation of $y$ and $u$ [16]. This problem, which has been unknown before, has caused performance loss and significant difficulties in applications (e.g. see [12], [13]) as LPV-IO models had been thought to be realizable as LPV-SS models according to the classical rules of the LTI realization theory. It has been demonstrated in [16] that using such an intuitive conversion between the two representation structures can lead to $40\%$ of output-error even for slowly varying $p$. Furthermore, in the air charge control problem of *Spark Ignition* (SI) gasoline engine, used in this paper for illustration, LPV controllers designed on the SS realization of an identified high validity LPV-IO model show a significant performance loss if the SS realization is obtained according to the LTI rules (see Section VI).

Since main-stream LPV controller synthesis approaches are based on state-space representations, obtaining SS realization of LPV-IO models has become an essential task to be solved in practice. According to a recently developed algebraic framework to give a solution for this transformation problem, see [7], [17], it has been proven that for obtaining equivalence between SS and IO representations, it is necessary to allow for a dynamic mapping between the scheduling signals and the system matrices (dynamic dependence), i.e. the system matrices must be allowed to depend on finite many time-shifted instances of $p(k)$, like $\{\ldots, p(k-1), p(k), p(k+1), \ldots\}$. This does increase the complexity of the produced SS model, which may prevent controller synthesis or hardware implementation of controller designs.

In this paper we propose practical and systematic methods to solve the problem of transforming LPV-IO models into LPV-SS forms by avoiding such dynamic dependence on the scheduling signals. Therefore, we assume that an identified and validated IO model of the system is given for which realization needs to be addressed, in other words, we deal here only with the realization problem. Hence, we do not intend to compare performance of identification via LPV-SS or IO approaches nor posing any of these model structures or methods to be superior above the other. The developed realization approaches propose a way to close the gap between LPV-IO modeling/identification and control synthesis. Additionally, a criterion is also established to decide when the LTI theory inspired realization can be used without introducing a significant approximation error. To reduce the order of the resulting SS realization an LPV Ho-Kalman type of model reduction approach is introduced, which can even reduce non-stable models. All ideas are illustrated with simulation studies on practical design examples.

The paper is formulated as follows: In Section II, the transformation problem between LPV-IO and LPV-SS representations is briefly discussed highlighting the underlaying difficulties and providing a general solution through an algebraic approach which requires the use of dynamic dependence. Next in Section III, a useful criterion is discussed to decide when the classical LTI realization theory can be applied without serious consequences and how much can be 'lost' in terms of model validity. In Section IV it is discussed in which special cases it is possible to avoid the introduction of dynamic dependence and to provide exact realization in terms of an LPV-SS model with static dependence. This is followed in Section V by a brief description of the LPV Ho-Kalman approach being able to reduce non-minimal SS models resulting from certain conversion methods. In Section VI, the performance of the proposed approaches is evaluated through modeling and control design for the intake manifold of a spark ignition gasoline engine. Finally, the conclusions of the paper are drawn in Section VII.

## II. THE TRANSFORMATION PROBLEM

In the conversion of LPV-IO models to LPV-SS representations it holds true in general that to preserve IO equivalence of the resulting descriptions, dynamic dependence of the coefficients on the scheduling signal $p$ (dependence on time-shifted version of $p$) must be considered [7], [17]. To illustrate the problem, investigate the following second-order SS representation:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & a_2(p(k)) \\ 1 & a_1(p(k)) \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} b_2(p(k)) \\ b_1(p(k)) \end{bmatrix} u(k),$$

$$y(k) = x_2(k).$$

With simple manipulations this system can be written in an equivalent IO form:

$$y(k) = a_1(p(k-1))y(k-1) + a_2(p(k-2))y(k-2) + b_1(p(k-1))u(k-1) + b_2(p(k-2))u(k-2),$$

which is clearly not in the form of (3) due to the dependence of the coefficients on $p(k-1)$ and $p(k-2)$. We can see from this example that it is necessary to allow for dynamic dependence of the varying parameters, like state-space matrices or IO coefficients, in order to characterize equivalent LPV-SS realizations of LPV-IO models. Next, we will investigate how we can reformulate our representations to handle such a dynamic dependence in a well-founded sense and how we can provide algorithms that characterize equivalent representations. For this purpose we will briefly introduce the algebraic framework of the so called LPV *behavioral approach* developed in [17].

### A. System representations

In order to describe the functional dependence of a single real-valued coefficient, we employ functions $r : \mathbb{R}^n \to \mathbb{R}$ that are considered to be in the field $\mathcal{R} = \cup_{n\in\mathbb{N}}\mathcal{R}_n$, where $\mathcal{R}_n$ is the set of essentially $n$-dimensional real-meromorphic functions (being the quotient of analytical real functions). In this context *essentially* means that $r(\mathtt{x}_1, \cdots, \mathtt{x}_n)$, where $\mathtt{x} = [\ \mathtt{x}_1 \ \ \ldots \ \ \mathtt{x}_n\ ] \in \mathbb{R}^n$, does depend on $\mathtt{x}_n$. The function $r$ specifies how a corresponding coefficient function (like $\{A, \ldots, D\}$ and $\{a_i, b_j\}$ in (1a-b) and (3), respectively) depends on $n$ variables, that are selected – in a unique ordering – from the set $\{q^i p_j\}_{j=1,\cdots,n_\mathrm{p}}^{i\in\mathbb{Z}}$. More specifically, for a given $\mathbb{P}$ with dimension $n_\mathrm{p}$ and $r \in \mathcal{R}_n$, label the variables $\mathtt{x}_1, \ldots, \mathtt{x}_n$ of $r$ as $\zeta_{0,1}, \zeta_{0,1}, \ldots$ according to the following ordering:

$$r(\zeta_{0,1}, \ldots, \zeta_{0,n_\mathrm{p}}, \zeta_{1,1}, \ldots, \zeta_{1,n_\mathrm{p}}, \zeta_{-1,1}, \ldots, \zeta_{-1,n_\mathrm{p}}, \zeta_{2,1}, \ldots).$$

For a given scheduling signal $p$, associate the variable $\zeta_{i,j}$ with $q^i p_j$. For this association we introduce the operator

$$\diamond : (\mathcal{R}, \mathbb{P}^\mathbb{Z}) \to \mathbb{R}^\mathbb{Z} \ \text{ defined by } \ r \diamond p = r\left(p, qp, q^{-1}p, \ldots\right),$$

where $\mathbb{X}^\mathbb{Z}$ stands for all maps from $\mathbb{Z}$ to $\mathbb{X}$. Thus the value of a ($p$-dependent) coefficient $r$ in an LPV system representation at time $k$ is given by $(r \diamond p)(k)$.

*Example 1 (Coefficient function):* Let $\mathbb{P} = \mathbb{R}^{n_\mathrm{p}}$ with $n_\mathrm{p} = 2$. Consider the real-meromorphic coefficient function $r : \mathbb{R}^3 \to \mathbb{R}$, defined as

$$r(\mathtt{x}_1, \mathtt{x}_2, \mathtt{x}_3) = \frac{1 + \mathtt{x}_3}{1 - \mathtt{x}_2}.$$

Then for a scheduling signal $p : \mathbb{Z} \to \mathbb{R}^2$:

$$(r \diamond p)(k) = r(p_1, p_2, qp_1)(k) = \frac{1 + p_1(k+1)}{1 - p_2(k)}.$$

On the other hand, if $n_\mathrm{p} = 3$, then $(r \diamond p)(k) = r(p_1, p_2, p_3)(k) = \frac{1+p_3(k)}{1-p_2(k)}$, showing that the operator $\diamond$ implicitly depends on $n_\mathrm{p}$.

In the sequel the (time-varying) coefficient sequence $(r \diamond p)$ will be used to operate on a signal $w$ (like $a_i(p)$ in (3)), giving the varying coefficient sequence of the representations. In this respect an important property of the $\diamond$ operation is that multiplication with the shift operator $q$ is not commutative, in other words

$$q(r \diamond p)w \neq (r \diamond p)qw. \tag{4}$$

To handle this multiplication, for $r \in \mathcal{R}$ we define the shift operations $\overrightarrow{r}, \overleftarrow{r}$ as

$$\begin{aligned} \overrightarrow{r} &= r' \in \mathcal{R} \quad \text{s.t.} \quad & r' \diamond p &= r \diamond (qp), \\ \overleftarrow{r} &= r'' \in \mathcal{R} \quad \text{s.t.} \quad & r'' \diamond p &= r \diamond (q^{-1}p), \end{aligned}$$

for all $p \in (\mathbb{R}^{n_\mathrm{p}})^{\mathbb{Z}}$. With these notions we can write $qr = \overrightarrow{r} q$ and $q^{-1}r = \overleftarrow{r} q^{-1}$ which corresponds to

$$q(r \diamond p)w = (\overrightarrow{r} \diamond p)qw \quad \text{and} \quad q^{-1}(r \diamond p)w = (\overleftarrow{r} \diamond p)q^{-1}w,$$

in the signal level. This non-commutativity of multiplication with $q$ in the LPV case is the core problem of realization.

*Example 2 (Shift operators):* Consider the coefficient function $r$ given in Example 1 with $n_\mathrm{p} = 2$. Then $\overrightarrow{r}$ is a function $\mathbb{R}^5 \to \mathbb{R}$, given by $\overrightarrow{r}(\zeta_{0,1}, \zeta_{0,2}, \zeta_{1,1}, \zeta_{1,2}, \zeta_{-1,1}, \zeta_{-1,2}, \zeta_{2,1}) = \frac{1+\zeta_{2,1}}{1-\zeta_{1,2}}$. For a scheduling trajectory $p : \mathbb{Z} \to \mathbb{R}^2$, it holds that $(\overrightarrow{r} \diamond p)(k) = (r \diamond (qp))(k) = \frac{1+p_1(k+2)}{1-p_2(k+1)}$.

Next we can introduce discrete-time LPV-IO and SS representations that have equivalent IO behavior. Let $\mathcal{R}[\xi]$ be the ring of polynomials in the indeterminate $\xi$ and with coefficients in $\mathcal{R}$. Since the indeterminate $\xi$ is associated with $q$, multiplication with $\xi$ is non-commutative on $\mathcal{R}[\xi]$, i.e. $\xi r = \overrightarrow{r}\xi$ and $r\xi = \xi\overleftarrow{r}$. Then for specified input and output variables $(y, u) \in (\mathbb{R}^{n_\mathbb{Y}} \times \mathbb{R}^{n_\mathbb{U}})^{\mathbb{Z}}$ of a given LPV system $\mathcal{S}$ we can introduce the *IO representation* of $\mathcal{S}$ as

$$(R_\mathrm{y}(q) \diamond p)\, y = (R_\mathrm{u}(q) \diamond p)\, u, \tag{5}$$

where $R_\mathrm{y} \in \mathcal{R}[\xi]^{n_\mathrm{y} \times n_\mathrm{y}}$ and $R_\mathrm{u} \in \mathcal{R}[\xi]^{n_\mathrm{y} \times n_\mathrm{u}}$ are matrix polynomials with meromorphic coefficients, e.g. $R_\mathrm{y}(q) = \sum_{i=0}^{n} a_i q^i$ where $a_i \in \mathcal{R}$, $R_\mathrm{y}$ is full rank and $\deg(R_\mathrm{y}) \geq \deg(R_\mathrm{u})$. It is apparent that (5) is the 'dynamic-dependent' counterpart of (3). Furthermore, we can characterize the solution space of (5) as all maps of $(y, u, p)$ with left-compact support that satisfy (5). We recognize (5) to be the representation of an LPV system $\mathcal{S}$ if its solution space contains all trajectories of $(y, u, p)$ that can happen during the operation of $\mathcal{S}$. Exact characterization of such behaviors even defining the conditions required for $\mathcal{S}$ to be an LPV system are given in [7], [17].

The natural counterpart of (5) to define SS representation of an LPV system $\mathcal{S}$ is

$$(R_\mathrm{w}(q) \diamond p)\mathrm{col}(u, y) = (R_\mathrm{L}(q) \diamond p)x, \tag{6}$$

where $\mathrm{col}(\centerdot)$ denotes the column-vector composition, $R_\mathrm{w} \in \mathcal{R}[\xi]^{n_\mathrm{r}\times(n_\mathrm{y}+n_\mathrm{u})}$ and $R_\mathrm{L} \in \mathcal{R}[\xi]^{n_\mathrm{r}\times n_\mathrm{x}}$ are matrix polynomials and $x$ is an auxiliary, so called *latent variable*. $x$ satisfies the property of a state variable if for every $k_0 \in \mathbb{Z}$ and each solutions $(x_1, y, u, p)$ and $(x_2, y, u, p)$ of (6) with $x_1(k_0) = x_2(k_0)$ it holds that concatenation of $(x_1, y, u, p)$ and $(x_2, y, u, p)$ at $k_0$ is a solution of (6). It can be shown that $x$ qualifies as a state variable in (6), if and only if (6) can be written in a form where $\deg(R_\mathrm{w}) = 0$ and $\deg(R_\mathrm{L}) = 1$ [17]. Then the state-space representation can be formulated as a first-order parameter-varying difference equation system in the state variable $x$ as

$$qx = (A \diamond p)x + (B \diamond p)u, \tag{7a}$$

$$y = (C \diamond p)x + (D \diamond p)u, \tag{7b}$$

with

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \in \left[\begin{array}{c|c} \mathcal{R}^{n_\mathrm{x}\times n_\mathrm{x}} & \mathcal{R}^{n_\mathrm{x}\times n_\mathrm{u}} \\ \hline \mathcal{R}^{n_\mathrm{y}\times n_\mathrm{x}} & \mathcal{R}^{n_\mathrm{y}\times n_\mathrm{u}} \end{array}\right]. \tag{8}$$

It is apparent that (7a-b) are the dynamic-dependent counterparts of (1a-b). It can be shown that (7a-b) is equivalent with (5) in the sense that for a given LPV-IO representation (5) there exists an LPV-SS representation (7a-b) with the same IO behavior. The latter means that under minor restrictions, for all trajectories of $(y, u, p)$ that satisfy (5) there exists a $x \in (\mathbb{R}^{n_\mathrm{x}})^\mathbb{Z}$ s.t. $(x, y, u, p)$ satisfies (7a-b).

*B. Equivalent state-space forms*

To define equality of LPV-SS and LPV-IO representations, we first have to clarify *state-transformations* in the LPV case. Consider an LPV-SS representation given by (7a-b). Let $T \in \mathcal{R}^{n_\mathrm{x}\times n_\mathrm{x}}$ be invertible (in $\mathcal{R}^{n_\mathrm{x}\times n_\mathrm{x}}$) and consider $x'$, given by

$$x' = (T \diamond p)x. \tag{9}$$

It is immediate that substitution of (9) into (7a) gives

$$q(T^{-1} \diamond p)x' = (A \diamond p)(T^{-1} \diamond p)x' + (B \diamond p)u. \tag{10}$$

Due to the fact that (10) is a first-order parameter varying difference equation w.r.t. $x'$, the latent varaible $x'$ trivially qualifies as a new state variable which yields that an equivalent LPV-SS representation of (7a-b) reads as

$$\left[\begin{array}{c|c} \overrightarrow{T} A T^{-1} & \overrightarrow{T} B \\ \hline C T^{-1} & D \end{array}\right]. \tag{11}$$

Similar to the LTI case it can be proven that two LPV state-space representations have the same IO behavior if and only if their state variables are related via a state-transformation (9). A major difference w.r.t. LTI state-transformations is that, in the LPV case, $T$ is not only a constant matrix but can be dependent on $p$ and this dependence can be dynamic, i.e. $T \in \mathcal{R}^{n_\mathrm{x} \times n_\mathrm{x}}$. Based on the developed state-transformation and the concept of state-observability and reachability matrices, the classical canonical forms can also be defined (see [7], [16]).

### *C. Input-output to state-space*

Now the results of [7], [17], which provide the basis for LPV input-output to state-space transformation, can be formulated as follows:

*Theorem 1:* Consider a given LPV-IO representation (5) and define $R(\xi) = [\ R_\mathrm{y}(\xi) \quad -R_\mathrm{u}(\xi)\ ]$. Assume that (5) is minimal in the sense that $R_\mathrm{y}(\xi)$ and $R_\mathrm{u}(\xi)$ are coprime. Let $w = \mathrm{col}(y, u)$ and select a full row rank $X \in \mathcal{R}[\xi]^{\cdot \times n_\mathrm{y}+n_\mathrm{u}}$ such that for

$$x = (X(q) \diamond p)w, \tag{12}$$

the latent variable $x$ satisfies the property of state, then there exist unique matrix functions $\{A, B, C, D\}$ in $\mathcal{R}^{\cdot \times \cdot}$ and polynomial matrix functions $X_\mathrm{u}, X_\mathrm{y} \in \mathcal{R}[\xi]^{\cdot \times \cdot}$ with appropriate dimensions such that

$$\xi X(\xi) = AX(\xi) + BS_\mathrm{u} + X_\mathrm{u}(\xi)R(\xi), \tag{13a}$$

$$S_\mathrm{y} = CX(\xi) + DS_\mathrm{u} + X_\mathrm{y}(\xi)R(\xi), \tag{13b}$$

where $S_\mathrm{u} \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{y}+n_\mathrm{u}}$ and $S_\mathrm{y} \in \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{y}+n_\mathrm{u}}$ are selector matrices[1] giving $u = S_\mathrm{u}w$ and $y = S_\mathrm{y}w$. For a proof see [17]. Note that $X$ in (12) can be generated from $R$ by using the so called *cut-and-shift operations* (see [18]). Note also that (13a-b) corresponds to a set of linear equations to be solved in order to obtain $\{A, B, C, D\}$ and $X_\mathrm{u}, X_\mathrm{y}$. With the resulting $\{A, B, C, D\}$, (8) is a minimal (in terms of state-dimension) state-representation of the LPV system $\mathcal{S}$. In the SISO case minimality is guaranteed for any choice of full row-rank $X$ satisfying (12), while in the MIMO case only appropriate selection strategies for $X$ lead to minimal realizations. Furthermore, specific choices of $X$ lead to specific canonical forms, see [17] for more details.

For a simple example consider an LPV-IO representation given in the form of

$$y = pq^{-1}y - pq^{-2}y + pq^{-2}u, \tag{14}$$

[1] A matrix with one entry 1 in each row, at most one entry 1 in each column, and all other entries 0 is a selector matrix.

with $\mathbb{P}=\mathbb{R}$. According to the above theorem, by applying (13a-b) with $(X(\xi)\diamond p)=\left[\begin{smallmatrix}\xi+qp & 0\\ 1 & 0\end{smallmatrix}\right]$, we obtain an LPV-SS realization of (14) in the form of

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]\diamond p=\left[\begin{array}{cc|c} 0 & q^2p & q^2p \\ 1 & -qp & 0 \\ \hline 0 & 1 & 0 \end{array}\right], \tag{15}$$

with $X_{\mathrm{u}}=\left[\begin{array}{cc} 1 & 0 \end{array}\right]^\top$ and $X_{\mathrm{y}}=0$. This LPV-SS form is the so called *companion-observability canonical* form of (14). From this example we can see, that via (13a-b) we can provide minimal SS realization of a given LPV-IO representation and in fact via state-transformations we can characterize all equivalent SS realizations of the system. However, using the results of Theorem 1 we have no control over the introduced scheduling dependence which is most likely to be dynamic and rational. As for the use of common LPV control synthesis tools most preferably we need realizations with simple static dependence like linear dependence on $p$, the basic question which rises is how we can arrive at a SS realization where also the scheduling dependence is “minimal”. So it is important to explore in which cases we can avoid the use of dynamic dependence, give direct realization forms and what price we must pay if we restrict ourselves to static dependence in terms of an approximative realization. These are the question we intend to address in the sequel.

## III. Criterion of dynamic dependence

In the literature, the issue of dynamic-dependence of the LPV representation on the scheduling parameters is often overlooked when an LPV-IO model is transformed into an LPV-SS representation, e.g. [13]. Instead, usually LTI realization theory is used to convert (3) to an LPV-SS form (1a-b) where the matrices have only static dependence. Based on this, (3) is commonly “realized” in terms of canonical forms, like the reachable (or so called companion reachability) form given in the SISO case as

$$\left[\begin{array}{c|c} A(p) & B(p) \\ \hline C(p) & D(p) \end{array}\right]=\left[\begin{array}{ccccc|c} -a_1(p) & -a_2(p) & \dots & -a_{n_\mathrm{a}-1}(p) & -a_{n_\mathrm{a}}(p) & 1 \\ 1 & 0 & \dots & \dots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \vdots \\ 0 & \dots & 0 & 1 & 0 & 0 \\ \hline c_1(p) & \dots & \dots & \dots & c_{n_\mathrm{a}}(p) & d(p) \end{array}\right], \tag{16}$$

where $d = b_0$ and $c_j = b_j - a_j b_0$ and if $n_\mathrm{b} < n_\mathrm{a}$, then $b_j = 0$ for $j > n_\mathrm{b}$. In [7], [16], it has been shown that in the equivalent reachable canonical form of (3), the coefficients have dynamic dependence on $p$ and they become rational functions of the original $a_i$ and $b_j$ coefficients of (3). This implies that (16) is at best an approximation of the true canonical realization of the LPV-IO model, and the introduced approximation error can indeed be arbitrary large [7].

However, dynamic dependence associated with a SS representation of the system commonly increases the complexity of control synthesis. Thus, it becomes a relevant question when this approximative realization can be used without serious performance degradation of the designed controller. A simple answer is to analyze the error between the true and the approximative realization. Building upon the realization theory derived in [7], it can be shown that the approximation error depends on how the Markov parameters $\{g_i\}_{i=0}^{\infty}$ of the plant are approximated. For a given trajectory of $p$ denote $h(p,k_0)(k)$ the impulse response of the system at time $k$ for an impulse applied to the system at $k_0$. Then $h(p,k_0)(k) = g_{k-k_0}(p,k)$ for $k \geq k_0$ and $h(p,k_0)(k) = 0$ for $k < k_0$. Note that in the LPV case the impulse response of the system depends on the trajectory of $p$ and the time instance when the impulse is applied. Considering the Markov parameters of (3) with $n_\mathrm{a} = n_\mathrm{b} = 1$ and $b_0(p) \neq 0$, it follows that for a given trajectory of $p$

$$\begin{aligned} g_0(p,k) &= b_0(p_k), \\ g_1(p,k) &= -a_1(p_k)b_0(p_{k-1}) + b_1(p_k), \\ g_2(p,k) &= -a_1(p_k)\big(b_1(p_{k-1}) - a_1(p_{k-1})b_0(p_{k-2})\big), \end{aligned} \tag{17}$$

etc., where $p_k = p(k)$ and it holds true that all solution trajectories of (3) with left compact support satisfy

$$y(k) = \sum_{l=0}^{\infty} g_l(p,k)u(k-l). \tag{18}$$

However, if the LTI realization theory is used, it is assumed that the Markov parameters are

$$\begin{aligned} \hat{g}_0(p,k) &= b_0(p_k), \\ \hat{g}_1(p,k) &= -a_1(p_k)b_0(p_k) + b_1(p_k), \\ \hat{g}_2(p,k) &= -a_1(p_k)\big(b_1(p_k) - a_1(p_k)b_0(p_k)\big), \end{aligned} \tag{19}$$

etc. Note the difference of time dependence for each Markov parameter $g_i$ and $\hat{g}_i$. As for order $n_\mathrm{a}$, the first $n_\mathrm{a}+1$ Markov parameters (with the feedthrough term) completely characterize the system dynamics in a functional sense, if

$$J = \sup_{p\in\mathbb{P}^{\mathbb{Z}}} \left\| \left[ g_0(p,k) \ \ \ldots \ \ g_{n_\mathrm{a}}(p,k) \right]^\top - \left[ \hat{g}_0(p,k) \ \ \ldots \ \ \hat{g}_{n_\mathrm{a}}(p,k) \right]^\top \right\|_\infty \tag{20}$$

is "small," then the worst-case difference between the IO behavior of the approximative and the true realization can be considered negligible. In (20), $\| \centerdot \|_\infty$ denotes the $\ell_\infty$ norm. However, in order to quantify what "small" or "acceptable" is from the viewpoint of the user or a particular application we can introduce a relative threshold $\epsilon > 0$ w.r.t. to the $\ell_\infty$ norm of the impulse response of the original Markov parameter sequence:

$$\bar{J} = \sup_{p\in\mathbb{P}^{\mathbb{Z}}} \left\| \left[ g_0(p,k) \ \ \ldots \ \ g_{n_\mathrm{a}}(p,k) \right]^\top \right\|_\infty. \tag{21}$$

In this sense, for a given $\epsilon > 0$ and $\bar{J} \neq 0$ we can consider the worst-case approximation error to be acceptable if

$$\frac{J}{\bar{J}} < \epsilon. \tag{22}$$

See Remark 1 for an interpretation of the error in this relative sense. Note that $J$ can be computed in practice by considering the supremum over the values of $g_0, \ldots, g_{n_\mathrm{a}}$ for finite sequences $[\ p(k) \ \ \ldots \ \ p(k-n_\mathrm{a})\ ] = [\ \mathtt{p}_0 \ \ \ldots \ \ \mathtt{p}_{n_\mathrm{a}}\ ] \in \mathbb{P}^{n_\mathrm{a}+1}$. Then by gridding of $\mathbb{P}^{n_\mathrm{a}+1}$ and assuming an upper bound on the rate of variation of $p$, i.e. $\|p(k)-p(k-1)\| < \eta$, approximate computation of $J$ becomes available in a lower bound sense. By assuming a set of trajectories $\mathcal{P} \subset \mathbb{P}^{\mathbb{Z}}$ of $p$ that are expected during the operation of the plant, the search space can even be further decreased.

*Example 3:* Consider an LPV-IO representation (3) with $n_\mathrm{a} = 9$, $n_\mathrm{b} = 2$, $\mathbb{P} = [-2\pi, 0]$ where the parameter dependent coefficients are given as follows:

$$\begin{array}{llll}
a_1(p) = 0.24 + 0.1p, & a_2(p) = 0.6 - 0.1\sqrt{-p}, & a_3(p) = 0.3\sin(p), & a_4(p) = 0.17 + 0.1p, \\
a_5(p) = 0.3\cos(p), & a_6(p) = -0.27, & a_7(p) = 0.01p, & a_8(p) = -0.07, \\
a_9(p) = 0.01\cos(p), & b_0(p) = 1, & b_1(p) = 1.25 - p, & b_2(p) = -0.2 - \sqrt{-p}.
\end{array}$$

Note that all coefficients have static dependence on $p$. A fine grid of $\mathbb{P}^{(n_\mathrm{a}+1)=10}$ is constructed, where each grid point represents a finite sequence of $p$ such that $\|p(k)-p(k-1)\| < \eta_1 = 0.01$. Then (20) and (21) are adopted to compute $J$ and $\bar{J}$ aiming for $\epsilon = 1\%$. The maximum of $J$ over the grid points is $J_1 = 0.054$, while $\bar{J} = 0.75$. Since $J_1 < \epsilon\bar{J}$, the LTI realization can be employed to convert this LPV-IO model with $\eta_1 = 0.01$ into an adequate LPV-SS form using for instance the reachable canonical form. To demonstrate, that satisfying criterion (20) indeed leads to a SS realization which meets with the expected quality in terms of approximation error, the best fit rate (BFR) or fit score (see [19])

$$\mathtt{BFR} = 100\%.\max\left(1 - \frac{\|y(k) - \hat{y}(k)\|_2}{\|y(k) - y_\mathrm{m}\|_2}, 0\right), \tag{23}$$

where $y_\mathrm{m}$ is the mean of $y$ and $\|\centerdot\|_2$ is the $\ell_2$ norm, is used. In other words, the BFR is applied to validate the obtained approximate LPV-SS model. At each grid point of $\mathbb{P}^{10}$, (23) is used to compute the error between the sequence of Markov parameters $g_0, \ldots, g_{n_\mathrm{a}}$ of the true LPV-IO model (3) and the sequence of Markov parameters $\breve{g}_0(p,k) = D(p_k), \ldots, \breve{g}_{n_\mathrm{a}}(p,k) = C(p_k)\prod_{i=1}^{n_\mathrm{a}-1} A(p_{k-i})B(p_{k-n_\mathrm{a}})$ of the LTI realization provided LPV-SS form with matrix functions $\{A, B, C, D\}$ having static dependence on $p$. Note that both $g_0, \ldots, g_{n_\mathrm{a}}$ and $\breve{g}_0, \ldots, \breve{g}_{n_\mathrm{a}}$ have dynamic dependence and $\{\breve{g}_i\}$ are not the same as in (19). The resulting worst case BFR over the grid points is $\mathsf{BFR}_1 = 96.73\%$. This means that using the LTI realization theory to construct an LPV-SS form of the original LPV-IO model leads to an acceptable worst-case approximation error $\mathsf{BFR}_1$ if $\|p(k) - p(k-1)\| < \eta_1$ is satisfied. The example can also be repeated with $\eta_2 = 0.3$. According to (20) the maximum achieved $J$ is $J_2 = 1.26$, so $J_2 \gg \epsilon\bar{J}$. Therefore the LTI realization concept in this case is not advised to be used as the resulting error can be considerable (larger than the specified 1%). This is proven by computing the worst-case $\mathsf{BFR}_2$ over the grid points which is only $67.39\%$ in this case.

In order to find a boundary $\bar{\eta}$ for which $J < \epsilon\bar{J}$, the following problem is solved

$$\bar{\eta} := \arg\inf_{\eta \geq 0}\ \epsilon\bar{J} - J \quad \text{ s.t. } \quad \epsilon\bar{J} - J > 0. \tag{24}$$

By solving (24) in this example, the resulting $\bar{\eta}$ is $0.0139$ which is the maximum allowed rate of change of $p$, in terms of (21), to guarantee that by applying the LTI realization theory concept on (3) the resulting SS form will approximate the IO behavior of (24) adequately (when $\epsilon$ is chosen as $1\%$). The worst case approximation error associated with $\bar{\eta}$ is BFR$= 96.27\%$ which can be considered as a good approximation of the original IO behavior of the LPV-IO model.

*Remark 1:* Note that the considered criterion is applicable both in the SISO and the MIMO cases. In the later case, the Markov parameters in (20), (21) are matrix functions and hence the infinity norm of a multidimensional signals is considered. Additionally, the criterion itself can be formulated in different ways. One can use the $\ell_2$ norm or different $\bar{J}$ according to the specific application or needs of the user, for instance, the induced $\mathcal{H}_\infty$ norm of the error between the two models. With respect to the latter case it can be shown that for an asymptotically stable LPV system, the induced $\mathcal{H}_\infty$ norm satisfies

$$\sup_{\substack{\|u\|_2<\infty \\ u\neq 0,\ p\in\mathbb{P}^{\mathbb{Z}}}} \frac{\|\sum_{i=0}^{\infty} g_i(p)q^{-i}u\|_2}{\|u\|_2} \leq \sup_{\substack{p\in\mathbb{P}^{\mathbb{Z}} \\ k\in\mathbb{Z}}} \|h(p,k)\|_2. \tag{25}$$

This shows that the criterion $J \leq \epsilon \bar{J}$ in terms of (20) with an $\ell_2$ norm and with an adequately large number of Markov parameters can be interpreted as bounding the $\mathcal{H}_\infty$ norm of the error in a relative sense. However, the error bound is delivered w.r.t. the assumption of static dependence of the Markov parameters and not directly the error between the LTI realization provided SS model and (3).

## IV. Dedicated LPV-SS realizations to ensure static dependence

By considering the realization theory briefly discussed in Section II and fully developed in [7], it can be shown that in special cases there exist ways of converting LPV-IO models to LPV-SS realizations without introducing dynamic dependence. However, each realization form is either based on specific assumptions or provides non-minimal SS realizations, which can be later converted into minimal realizations using appropriate tools, see Section V.

### *A. Shifted form*

For the sake of simplicity we consider only the SISO case. However, the realization forms that will be introduced can be extended to the MIMO case in a straightforward manner similar to the LTI case. Assume that the LPV-IO model is given in the form

$$y + \sum_{i=1}^{n_\mathrm{a}} a_i(q^{-i}p)q^{-i}y = \sum_{j=0}^{n_\mathrm{b}} b_j(q^{-j}p)q^{-j}u, \tag{26}$$

where $a_i, b_j : \mathbb{P} \rightarrow \mathbb{R}$. Note that $a_i$ and $b_j$ have a special form of dynamic dependence which can be introduced into the parametrization of most of the available LPV-IO identification approaches. Now introduce $x_1$ as

$$y = x_1 + b_0(p)u, \tag{27a}$$

$$qx_1 = \sum_{j=0}^{n_\mathrm{b}-1} b_{j+1}(q^{-j}p)q^{-j}u - \sum_{i=0}^{n_\mathrm{a}-1} a_{i+1}(q^{-i}p)q^{-i}y. \tag{27b}$$

Continue the so called natural state construction (see [7]) as

$$qx_1 = x_2 - a_1(p)y + b_1(p)u, \tag{28a}$$

$$qx_2 = \sum_{j=0}^{n_\mathrm{b}-2} b_{j+2}(q^{-j}p)q^{-j}u - \sum_{i=0}^{n_\mathrm{a}-2} a_{i+2}(q^{-i}p)q^{-i}y. \tag{28b}$$

until $qx_{n_a} = -a_{n_a}(p)y + b_{n_a}(p)u$. Thus we obtain a set of first-order difference equations:

$$q \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_{n_a-1} \\ x_{n_a} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & 1 \\ 0 & \dots & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_{n_a-1} \\ x_{n_a} \end{bmatrix} + \begin{bmatrix} -a_1(p) & b_1(p) \\ \vdots & \vdots \\ \vdots & \vdots \\ -a_{n_a-1}(p) & b_{n_a-1}(p) \\ -a_{n_a}(p) & b_{n_a}(p) \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix}. \tag{29}$$

Then by using (27a), $y$ can be eliminated from (29) yielding the LPV-SS representation:

$$\left[\begin{array}{c|c} A(p) & B(p) \\ \hline C(p) & D(p) \end{array}\right] = \left[\begin{array}{ccccc|c} -a_1(p) & 1 & 0 & \dots & 0 & b_1(p) - a_1(p)b_0(p) \\ \vdots & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ -a_{n_a-1}(p) & 0 & \dots & 0 & 1 & b_{n_a-1}(p) - a_{n_a-1}(p)b_0(p) \\ -a_{n_a}(p) & 0 & \dots & \dots & 0 & b_{n_a}(p) - a_{n_a}(p)b_0(p) \\ \hline 1 & 0 & \dots & \dots & 0 & d(p) \end{array}\right], \tag{30}$$

where $d = b_0$. This SS realization is minimal in the SISO case and has only static dependence. In the MIMO case, by computing a set of maximally independent rows of $[A, B]$ in (30), which corresponds to independent state-variables, minimal realization via this approach also follows.

### *B. Augmented SS form*

Assume that the LPV-IO model is given in the form

$$y + \sum_{i=1}^{n_a} a_i(q^{-1}p)q^{-i}y = \sum_{j=1}^{n_b} b_j(q^{-1}p)q^{-j}u. \tag{31}$$

Note that $a_i$ and $b_j$ have a special form of dynamic dependence which again can be introduced into the parametrization of most of the available LPV-IO identification approaches. It is also important that there is no feedthrough term, i.e. $b_0 = 0$. Under these conditions, an augmented equivalent LPV-SS representation is given in a straight forward manner as (32). Note that (32) is not minimal, but has only coefficients with static dependence. This type of augmented SS form is widely known in the LTI literature and also used in the identification toolbox in MATLAB [20]. As a next step, an LPV model reduction algorithm, see Section V, can be applied to reduce the state dimension, but preserve the static dependence. Based on Section IV-D, it holds true that the minimal state dimension allowing static dependence of the SS form is larger than or equal to the actual order of the system. Moreover, if the original IO model is already identified in the form of (3), then this approach is still applicable as the resulting LPV-SS realization will only be dependent on $qp$. If $p$ is measurable by a faster sampling rate than $u$ and $y$, or the variation of $p$ is much slower than the sampling rate of $u$ and $y$, then dependence on $qp$ does not introduce complexity into the model in terms of control design.

$$\left[\begin{array}{c|c} A(p) & B(p) \\ \hline C(p) & D(p) \end{array}\right] = \left[\begin{array}{cccccc|c} -a_1(p) & \dots & -a_{n_\mathrm{a}}(p) & b_2(p) & \dots & b_{n_\mathrm{b}}(p) & b_1(p) \\ 1 & 0 & \dots & \dots & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 1 \\ 0 & \dots & 0 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 1 & 0 \\ \hline 1 & 0 & \dots & \dots & \dots & 0 & 0 \end{array}\right]. \tag{32}$$

## *C. Observability form*

Suppose that the LPV-IO model is given in the form

$$y + \sum_{i=1}^{n_\mathrm{a}} a_i(q^{-n_\mathrm{a}}p)q^{-i}y = b_{n_\mathrm{a}}(q^{-n_\mathrm{a}}p)q^{-n_\mathrm{a}}u. \tag{33}$$

Note that $a_i$ and $b_j$ have a special form of dynamic dependence and the input has a delay of $q^{-n_\mathrm{a}}$. Based on the realization theory of observability canonical forms (see [7]), an equivalent SS realization reads as (34). Note that this LPV-SS realization is minimal in the SISO case and has only static dependence. In the MIMO case, a minimal form of (34) can be computed via finding the basis of its observability matrix following Young's selection scheme II and computing a state-transformation to obtain a MIMO observability canonical form, see [7, Ch.4].

$$\left[\begin{array}{c|c} A(p) & B(p) \\ \hline C(p) & D(p) \end{array}\right] = \left[\begin{array}{ccccc|c} 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 & 0 \\ -a_{n_\mathrm{a}}(p) & -a_{n_\mathrm{a}-1}(p) & \dots & \dots & -a_1(p) & b_{n_\mathrm{a}}(p) \\ \hline 1 & 0 & \dots & \dots & 0 & 0 \end{array}\right]. \tag{34}$$

### D. Elimination of dynamic dependence via state-transformation

In case the previous schemes can not be applied, the realization approach (13a-b) in Theorem 1 can be used, followed by a search for a state transformation which simplifies the coefficient dependence on $p$ to a static relation. In this way, it can be clearly characterized what is the minimal achievable complexity of the LPV-SS realization of the given model. Such an approach is tempting from the theoretical point of view but as we will see it has a significant computational load.

In order to avoid detailed mathematical treatment of the problem, we consider the above given idea in a simple case. For a given SISO LPV-IO representation (5) with $n_\mathrm{a} = 2$ and $0 \leq n_\mathrm{b} \leq n_\mathrm{a}$, an LPV-SS realization of (5) is obtained in the form of

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \diamond p = \left[\begin{array}{cc|c} \alpha_{11} & \alpha_{12} & \beta_1 \\ \alpha_{21} & \alpha_{22} & \beta_2 \\ \hline \gamma_1 & \gamma_2 & \delta \end{array}\right] \diamond p \tag{35}$$

where the coefficients $\alpha_{11}, \ldots, \delta \in \mathcal{R}$ can have dynamic dependence on $p$. Note that in terms of (11), no state transformation can annihilate dynamic dependence in $D$, which implies that a SS equivalent of (35) exists only in the case if $\delta$ has static dependence. The latter condition for the realization of an LPV-IO model in the form of (3) is automatically satisfied. Consider

$$M_1 = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \quad M_2 = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix}, \tag{36}$$

being full rank matrix functions in $\mathcal{R}^{2\times 2}$ such that

$$\bar{A}(p) = (M_1 A M_2) \diamond p, \qquad \bar{B}(p) = (M_1 B) \diamond p, \tag{37a}$$

$$\bar{C}(p) = (C M_2) \diamond p, \qquad I = (\overleftarrow{M}_1 M_2) \diamond p, \tag{37b}$$

where $\bar{A}$, $\bar{B}$, $\bar{C}$ are matrix functions with static $p$-dependence and $\overleftarrow{\cdot}$ means backward time-shift in the coefficient dependence like $\overleftarrow{M}_1 \diamond p = M_1 \diamond (q^{-1}p)$. The conditions (37a-b) imply the existence of a state-transformation such that the resulting SS form with matrices $\{\bar{A}, \bar{B}, \bar{C}, D\}$ has only static dependence. Furthermore, (37a-b) corresponds to a system of bilinear equations. These bilinear equations can be symbolically solved if it is assumed that $A$, $B$, $C$ are given in a canonical form. Consider the example in Section II-C where we developed an LPV-SS realization (15) via (13a-b) of the LPV-IO model (14). By solving (37a-b) w.r.t. (15) we can arrive at the equivalent LPV-SS representation

$$\left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & D \end{array}\right] \diamond p = \left[\begin{array}{cc|c} -p & 1 & 0 \\ p & 0 & 1 \\ \hline p & 0 & 0 \end{array}\right] \quad \text{with} \quad M_1 = \begin{bmatrix} 0 & \frac{1}{qp} \\ \frac{1}{q^2 p} & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & qp \\ p & 0 \end{bmatrix}, \tag{38}$$

where the state-space matrices have only static dependence. This demonstrates that in some cases it is possible to eliminate the dynamic dependence by using simple state-transformations.

In case no solution exists for (37a-b), it means that no minimal LPV-SS realization of the model exists with static dependence. In this situation,

$$M_1 = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \\ m_{31} & m_{32} \end{bmatrix}, \quad M_2 = \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{21} & \mu_{22} & \mu_{23} \end{bmatrix}, \tag{39}$$

can be considered where $M_1$ is full column and $M_2$ is full row rank, increasing the freedom of the coefficient transformations by introducing an extra state variable. Then, solvability can be checked again. Note that due to the increased number of state-variables, uniqueness of the solution (if exists) in terms of $M_1$ and $M_2$ is not guaranteed. In case of no solution, the augmentation can be continued till feasibility or when the number of state variables exceeds a limit.

*Proposition 1:* Consider (3) and assume that an LPV-SS realization of (3) is given in the form of (7a-b) with dynamic dependence. If no state-transformation can be found that leads to an equivalent SS form of (7a-b) with $2n_\mathrm{a}$ state-variables and with only static dependence, then an exact SS realization of (3) with static dependence is not available.

The proof of the above given proposition is based on the fact that SS realization of (3) can only introduce maximum $n_\mathrm{a}$-order of time shifts in the dependence of the coefficients. If this order of dynamic dependence can not be annihilated via $n_\mathrm{a}$ extra state variables, then it is also not possible by using $n_\mathrm{a}+1$ state variables, hence an equivalent SS realization with static dependence does not exist for (3).

It is an important remark that in general it is absolutely not guaranteed that an arbitrary LPV-IO model has an SS realization with static dependence. Moreover, symbolic solution of the state-transformation problem is a computationally demanding operation with an exponentially increasing memory and computational load. Therefore, in terms of practical use, elimination of dynamic dependence via state transformation is limited to small scale problems.

## V. LPV MODEL ORDER REDUCTION

As we could see in the previous section, in some cases it is possible to obtain an equivalent LPV-SS realization of a given LPV-IO model such that the resulting state matrices have only static dependence. However, in some of these situations, the price we need to pay for such a realization is the introduction of extra state variables. Often an approximation of such LPV-SS models with a more economical size of state dimension is required in order to decrease the complexity of control synthesis. So the natural question that rises is how we can efficiently apply model reduction of the resulting realizations in order to eliminate non-significant dynamics if needed.

### *A. Overview of LPV model reduction*

Model reduction of LPV systems has been already studied in the literature since the middle 1990s and both exact and approximative reduction techniques have been developed. Exact reduction techniques, such as [21], aim at a Kalman type of decomposition of the model to a minimal and an eliminatable part preserving IO equivalence. However, in case of identified LPV models, possibly with some over-parametrization, such decomposition is ill-posed due to the effect of noise on the estimates. On the other hand, some of the discussed realization approaches introduce extra-states to achieve static-dependence of the resulting LPV-SS form, hence such states are not eliminatable without using dynamic dependence. This implies that often only approximative methods can be used to obtain an economical model order in the considered problem setting. In terms of approximative approaches, LTI reduction techniques, such as coprime factor [22], optimal Hankel norm and balanced truncation [23] methods, are extended to LPV systems and some of these approaches are implemented in the Enhanced LFT toolbox [24], [25]. Model reduction based on balanced truncation is a well grounded scheme in theory and most often used [26]. However, most common and practical LPV model reduction techniques that are based on balanced truncation require quadratic stabilizability and detectability of the full order model and they can not easily deal with unstable models [23]. Both are important restrictions as quadratic stabilizability and detectability of an LPV model are not always guaranteed and a major application area of system identification is to obtain models of general unstable plants operated in closed loop. In the LTI case, the basic model reduction methods in combination with coprime factorization [27] or spectral decomposition techniques [28], can be used to reduce unstable systems. However extending such methods to LPV systems is accompanied by a significant increase of the complexity of the reduction technique [22]. Thus, conventional model reduction approaches are often not applicable to reduce the state-dimension of the SS realization of identified LPV models. Next, we investigate an alternative model reduction method to overcome these restrictions.

### *B. Hankel matrix*

In the sequel, we introduce a model reduction approach for MIMO LPV-SS representations with affine dependence on $p$. The proposed approach is an extension of the Ho-Kalman realization algorithm well known in the LTI case, see e.g. [29], and it is different from optimal Hankel norm and balanced truncation approaches [23], [24]. The method is proposed as a tool to assist in obtaining an economical-size LPV-SS realization of identified LPV-IO models. Unlike other LPV model reduction techniques, the proposed technique here, in addition to its simplicity, does not necessitate quadratic stabilizability or detectability of the full order model and it can be employed for both stable and unstable systems without imposing any modification.

Consider the LPV-SS representation (1a-b), with affine dependence of the system matrices:

$$A(p) = A_0 + \sum_{i=1}^{n_\psi} A_i \psi_i(p), \qquad B(p) = B_0 + \sum_{i=1}^{n_\psi} B_i \psi_i(p), \tag{40a}$$

$$C(p) = C_0 + \sum_{i=1}^{n_\psi} C_i \psi_i(p), \qquad D(p) = 0, \tag{40b}$$

where $\psi_i(\centerdot) : \mathbb{P} \to \mathbb{R}$ are analytic functions on $\mathbb{P}$ and $\{A_i, B_i, C_i\}_{i=0}^{n_\psi}$ are constant matrices with appropriate dimensions. Furthermore, for well-possedness it is assumed that $\{\psi_i\}_{i=1}^{n_\psi}$ are linearly independent on $\mathbb{P}$ and normalized w.r.t. an appropriate norm or inner product. Define

$$\mathcal{M}_1 = [\ B_0 \quad \dots \quad B_{n_\psi}\ ], \qquad \mathcal{M}_j = [\ A_0 \mathcal{M}_{j-1} \quad \dots \quad A_{n_\psi} \mathcal{M}_{j-1}\ ]. \tag{41}$$

Inspired by [30], [31], introduce the so called $k$-step extended reachability matrix of (1a-b) as

$$\mathcal{R}_k = [\ \mathcal{M}_1 \quad \dots \quad \mathcal{M}_k\ ], \tag{42}$$

where $\mathcal{R}_k \in \mathbb{R}^{n_\mathrm{x} \times \left(n_\mathrm{u} \sum_{l=1}^{k} (1+n_\psi)^l\right)}$. The matrix $\mathcal{R}_k$ is called the extended reachability matrix, as similar to the LTI case, full rank of $\mathcal{R}_{n_\mathbb{X}}$ is a necessary condition for reachability of (1a-b), however, opposite to the LTI case, it is not a sufficient condition [7], [32]. Let

$$P_k = [\ 1 \quad \psi_1(p(k)) \quad \dots \quad \psi_{n_\psi}(p(k))\ ]^\top, \tag{43a}$$

$$K_{k|i} = P_k \otimes \ldots \otimes P_{k-i} \otimes I_{n_\mathrm{u}}, \tag{43b}$$

$$M_{ki} = \mathrm{diag}(K_{k-i|0}, \ldots, K_{k-1|i-1}), \tag{43c}$$

where $\otimes$ stands for the Kronecker product and $I_n$ is the identity matrix with dimension $n$. Now it is true that if $x(k-i) = 0$ with $k > i \geq 0$, then

$$x(k) = \mathcal{R}_i M_{ki} U_{ki}, \tag{44}$$

where $U_{ki} = [\ u^\top(k-1) \quad \dots \quad u^\top(k-i)\ ]^\top$. From (44) it is obvious that for reachability, full-rank of $\mathcal{R}_{n_\mathbb{X}} M_{kn_\mathbb{X}}$ is required for all $k$ and every possible trajectory of $p$. Hence full-rank of $\mathcal{R}_{n_\mathbb{X}}$ corresponds to reachability only in a structural sense. Now, define

$$\mathcal{N}_1 = [\ C_0^\top \quad \dots \quad C_{n_\psi}^\top\ ]^\top, \qquad \mathcal{N}_j = [\ (\mathcal{N}_{j-1}A_0)^\top \quad \dots \quad (\mathcal{N}_{j-1}A_{n_\psi})^\top\ ]^\top, \tag{45a}$$

$$L_{k|i} = I_{n_\mathrm{y}} \otimes (P_k \otimes \cdots \otimes P_{k-i})^\top, \tag{45b}$$

$$N_{ki} = \mathrm{diag}(L_{k-i|0}, \ldots, L_{k-1|i-1}). \tag{45c}$$

Based on similar considerations as before, introduce the $k$-step extended observability matrix as

$$\mathcal{O}_k = [\ \mathcal{N}_1^\top \quad \dots \quad \mathcal{N}_k^\top\ ]^\top \tag{46}$$

where $\mathcal{O}_k \in \mathbb{R}^{\left(n_\mathrm{y} \sum_{l=1}^{k}(1+n_\psi)^l\right) \times n_\mathrm{x}}$. Now it is true for $k > i \geq 0$ that

$$Y_{ki} = N_{ki} \mathcal{O}_i x(k-i), \tag{47}$$

where $Y_{ki} = [\ y^\top(k-i) \quad \dots \quad y^\top(k-1)\ ]^\top$. Therefore,

$$\mathcal{H}_{ij} = \mathcal{O}_i \mathcal{R}_j \in \mathbb{R}^{\left(n_\mathrm{y} \sum_{l=1}^{i}(1+n_\psi)^l\right) \times \left(n_\mathrm{u} \sum_{l=1}^{j}(1+n_\psi)^l\right)}, \tag{48}$$

can be considered as the extended Hankel matrix of (1a-b). If $n_\mathbb{P} = 1$, then

$$\mathcal{H}_{ij} = \begin{bmatrix} C_0B_0 & C_0B_1 & C_0A_0B_0 & C_0A_0B_1 & \dots \\ C_1B_0 & C_1B_1 & C_1A_0B_0 & C_1A_0B_1 & \dots \\ C_0A_0B_0 & C_0A_0B_1 & C_0A_0^2B_0 & C_0A_0^2B_1 & \dots \\ C_1A_0B_0 & C_1A_0B_1 & C_1A_0^2B_0 & C_1A_0^2B_1 & \dots \\ C_0A_1B_0 & C_0A_1B_1 & C_0A_1A_0B_0 & C_0A_1A_0B_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

For sufficiently large $i$ and $j$, $\mathrm{rank}(\mathcal{H}_{ij}) = n$, which can be considered as the McMillan degree of (1a-b), in case of a minimal representation $n = n_\mathrm{x}$.

### C. LPV Ho-Kalman algorithm

To construct a minimal state-space realization of (1a) or (32) with static dependence we can proceed as follows. A Hankel matrix of the representation for sufficiently large dimensions is constructed such that $\mathrm{rank}(\mathcal{H}_{ij}) = n$ holds true. Then for any (full rank) matrix decomposition:

$$\mathcal{H}_{ij} = H_1 H_2, \tag{49}$$

with constant matrices $H_1 \in \mathbb{R}^{(n_\mathrm{y} \sum_{l=1}^{i}(1+n_\psi)^l)\times n}$ and $H_2 \in \mathbb{R}^{n\times(n_\mathrm{u} \sum_{l=1}^{j}(1+n_\psi)^l)}$ satisfying $\mathrm{rank}(H_1) = \mathrm{rank}(H_2) = n$, there exist matrices functions $\hat{A}(p), \hat{B}(p), \hat{C}(p)$ defined as in (40a-b), such that the $i$-step observability matrix $\hat{\mathcal{O}}_i$ and the $j$-step reachability matrix $\hat{\mathcal{R}}_j$ generated from their constant matrices satisfy

$$H_1 = \hat{\mathcal{O}}_i, \qquad H_2 = \hat{\mathcal{R}}_j. \tag{50}$$

The matrices $\hat{A}(p), \hat{B}(p), \hat{C}(p)$ can be computed as follows: When given $H_1 = \hat{\mathcal{O}}_i$, the matrices $[\ \hat{C}_0^\top \ \ \ldots \ \ \hat{C}_{n_\psi}^\top \ ]^\top$ can be extracted by taking the first $n_\mathrm{y}(1+n_\psi)$ rows and when given $H_2 = \hat{\mathcal{R}}_i$, the matrices $[\ \hat{B}_0 \ \ \ldots \ \ \hat{B}_{n_\psi} \ ]$ can be extracted by taking the first $n_\mathrm{u}(1+n_\psi)$ columns. The matrices $\{A_0, \ldots, A_{n_\psi}\}$ can be isolated by using a shifted Hankel matrix $\overleftarrow{\mathcal{H}}_{ij}$, which is simply obtained from the original Hankel matrix $\mathcal{H}_{ij}$, by shifting the matrix one block column, i.e. $n_\mathrm{u}(1+n_\psi)$ columns, to the left. It can be directly verified that $\overleftarrow{\mathcal{H}}_{ij}$ can be written as

$$\overleftarrow{\mathcal{H}}_{ij} = \mathcal{O}_i[\ A_0 \ \ \ldots \ \ A_{n_\psi} \ ](I_{1+n_\psi} \otimes \mathcal{R}_{j-1}), \tag{51a}$$

$$= H_1[\ \hat{A}_0 \ \ \ldots \ \ \hat{A}_{n_\psi} \ ](I_{1+n_\psi} \otimes \hat{H}_2), \tag{51b}$$

where $\hat{H}_2$ is generated from $H_2$ by leaving out the last $n_\mathrm{u}(1+n_\psi)^j$ columns. Under the assumption that $j$ is large enough that $\mathrm{rank}(\hat{H}_2) = n$, there exist pseudo inverse matrices $H_1^\dagger$ and $\hat{H}_2^\dagger$, such that

$$H_1^\dagger H_1 = I, \qquad \hat{H}_2 \hat{H}_2^\dagger = I. \tag{52}$$

These matrices can be computed as $H_1^\dagger = (H_1^\top H_1)^{-1} H_1^\top$ and $\hat{H}_2^\dagger = \hat{H}_2^\top (\hat{H}_2^\top \hat{H}_2)^{-1}$. As a consequence, it holds that

$$H_1^\dagger \overleftarrow{\mathcal{H}}_{ij}(I_{1+n_\psi} \otimes \hat{H}_2^\dagger) = [\ \hat{A}_0 \ \ \ldots \ \ \hat{A}_{n_\psi} \ ]. \tag{53}$$

A reliable procedure to compute the full rank decomposition of $\mathcal{H}_{ij}$ is to use the *singular value decomposition* (SVD):

$$\mathcal{H}_{ij} = U_k \Sigma_k V_k^\top, \tag{54}$$

where $U_k$ and $V_k$ are unitary matrices and $\Sigma_k$ is a diagonal matrix with positive entries $\sigma_1 \geq \cdots \geq \sigma_k$ referred to as singular values. The choice $H_1 = U_k \Sigma_k^{\frac{1}{2}}$, $H_2 = \Sigma_k^{\frac{1}{2}} V_k^\top$ directly leads to $H_1^\dagger = \Sigma_k^{-\frac{1}{2}} U_k^\top$, $H_2^\dagger = V_k \Sigma_k^{-\frac{1}{2}}$. By truncating the resulting $H_2$ to generate $\hat{H}_2$ and $\hat{H}_2^\dagger$, the matrices $\{\hat{A}_0, \ldots, \hat{A}_{n_\psi}\}$ can be calculated according to (53).

As in the LTI case, it is demonstrated here that the LPV Ho-Kalman algorithm can be applied to reduce the order or to find a minimal realization with a finite number of Markov parameters, provided that certain rank conditions on a finite Hankel matrix are satisfied. Furthermore for asymptotically stable systems, it can be shown that the state-space model obtained using the LPV Ho-Kalman algorithm with SVD is balanced, i.e. the controllability and the observability gramians of the state-space representation are equal, see [29] for more details. Theoretical error bounds to characterize the approximation error of the reduced models has not been developed yet, remaining to be the objectives of future research on this approach. However, numerical approximation of the error between the reduced and the original plant is computable in a $\mu$-test sense with the Enhanced LFT toolbox [24], [25]. An additional property of the algorithm is that with increasing $i$ and $j$, the size of $\mathcal{H}_{ij}$ exponentially increases while increasing $n_\psi$ results in a polynomial growth of $\mathcal{H}_{ij}$. This means that numerical problems and memory limits can quickly play a limiting factor if the algorithm is applied for large scale models. Additional numerical problems can also arise in case of unstable systems when too high powers of $\{\hat{A}_0, \ldots, \hat{A}_{n_\psi}\}$ can result in a badly conditioned Hankel matrix. The curse of dimensionality, which is a drawback of the algorithm, and the associated numerical problems can be reduced by applying a kernel based approach like in [31], numerical regularization, block-SVD algorithms, or Krylov subspace methods [33].

*Example 4:* Consider an LPV-IO model in the form of (31) with $n_\mathrm{a} = 3$, $n_\mathrm{b} = 2$, $\mathbb{P} = [-2\pi, 0]$ and

$$a_1(q^{-1}p) = (0.24 + 0.1 q^{-1}p), \qquad a_2(q^{-1}p) = (0.6 - 0.1\sqrt{-q^{-1}p}), \tag{55a}$$

$$a_3(q^{-1}p) = 0.3\sin(q^{-1}p), \qquad b_1(q^{-1}p) = (1.25 - q^{-1}p), \tag{55b}$$

$$b_2(q^{-1}p) = -(0.2 + \sqrt{-q^{-1}p}), \tag{55c}$$

By calculating the augmented SS form, a 4$^{\text{th}}$ order non-minimal LPV-SS representation results as shown in (32). To compute a lower order representation of the produced LPV-SS model, the LPV Ho-Kalman algorithm is applied.

According to (55a-b), the state matrices of the obtained LPV-SS model are written in terms of (40a-d) with $n_\psi = 3$, $\psi_1(p) = p$, $\psi_2(p) = \sqrt{-p}$ and $\psi_3(p) = \sin(p)$. Next the 3-step extended reachability $\mathcal{R}_3$ as well as the 3-step extended observability $\mathcal{O}_3$ matrices are constructed in terms of (42) and (46), respectively. These matrices are sufficient to perform model reduction by the introduced algorithm. Continuing, the extended Hankel matrix $\mathcal{H}_{33}$, see (48), is computed and an SVD is performed to decompose the Hankel matrix into $H_1, H_2$ and to obtain their pseudo inverses as in (49), (54). The singular values of the diagonal matrix $\Sigma_3$, see (48), are $[2.6253,\ 1.3759,\ 1.0136,\ 0.382,\ 0,\ \ldots,\ 0]$. It is clear that the fourth singular value is less significant than the rest, which suggests that it is possible to approximate the model with a third order one. Next, the shifted Hankel matrix $\overleftarrow{\mathcal{H}}_{33}$ is determined and the matrix $\hat{H}_2$ is generated. Finally the matrices $\{\hat{B}_0, \ldots, \hat{B}_{n_\psi}, \hat{A}_0, \ldots, \hat{A}_{n_\psi}, \hat{C}_0, \ldots, \hat{C}_{n_\psi}\}$ can be calculated using (53). Here $\hat{A}_i \in \mathbb{R}^{3\times 3}, \hat{B}_i \in \mathbb{R}^{3\times 1}, \hat{C}_i \in \mathbb{R}^{1\times 3}, i = 1, \ldots n_\psi$. 3D plots of the *Frozen Frequency Response* (FFR) of the LPV-IO model (frequency response of the model for constant $p(k)$, i.e. $p(k) =$ p for all $k \in \mathbb{Z}$.) and the difference between it and the reduced model for all values of $\mathbb{P}$ are shown in Fig. 1. The approximate LPV-SS model preserves the static dependence on the scheduling signal with the same scheduling functions, furthermore it gives balanced realization with acceptable representation error (see Fig. 1). In addition, the reduced LPV-SS model is compared with the LPV-IO model in terms of the BFR of simulated output over a set of 500 random trajectories of $u$ and $p$ with arbitrary fast variation of $p$. The minimum and the maximum values of BFR were $84.21\%$ and $94.75\%$, respectively. This justifies that the obtained reduced model via the LPV Ho-Kalman algorithm gives an acceptable approximation of the original system.

## VI. Simulation example (charge control)

In this section, the methods discussed in Sections IV and V are tested and compared on an application-based simulation study. The example considered here is the air charge control problem of a *spark ignition* (SI) engine.

### A. The engine manifold

The intake manifold of a SI engine for air charge control has a highly nonlinear nature. It is not an isolated system but part of the overall car model, Fig. 2. The opening of the throttle valve in the intake manifold $\alpha_{\mathrm{lim}}$ is used to control the amount of the normalized air charge $m_{\mathrm{nac}}$. The speed of the vehicle $p_2$ influences the internal dynamics of the intake manifold and the engine

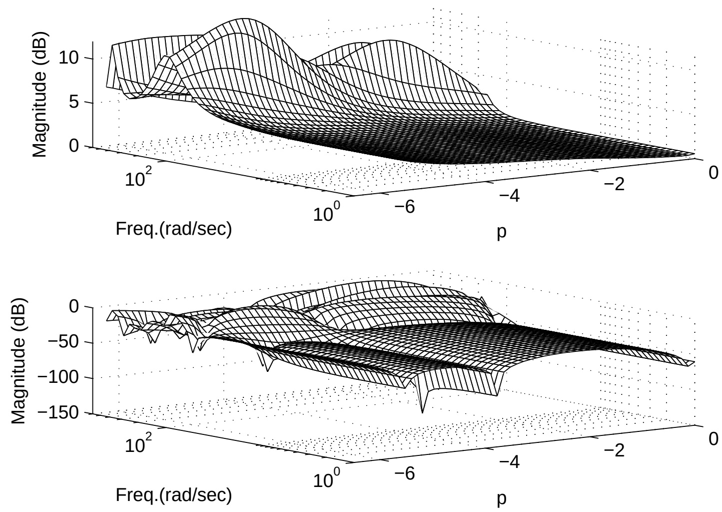



Fig. 1. 3D plot of the frozen frequency response functions of the LPV-IO model for all values in $\mathbb{P}$ (top) and the difference between these responses and the frozen frequency response functions of the reduced model (bottom) (Example 4).

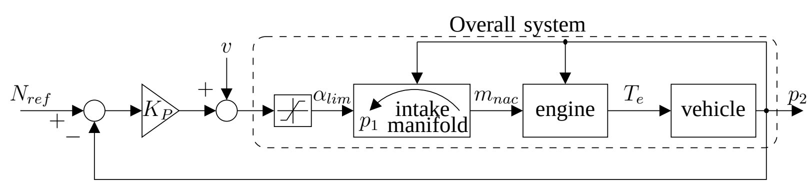



Fig. 2. Overall system including the engine and the vehicle in a closed-loop setting.

itself. The vehicle model, shown in Fig. 2, has an integral behavior, thus a negative feedback is applied between $p_2$ and $\alpha_{\mathrm{lim}}$ through a proportional gain $K_{\mathrm{p}} = \frac{3}{110}$ in order to stabilize the engine speed.

A parameterized nonlinear physical model of the overall process was provided and experimentally validated by the IAV GmbH company, Gifhorn, Germany; see [34] for more details. Constructing an LPV model from the physical model has two drawbacks:

1) It necessitates approximation of some nonlinear characteristics in an ad-hoc manner which reduces the accuracy of the derived LPV model.
2) It provides a continuous-time LPV model, and digital implementation of a controller designed in continuous-time leads to performance deterioration due to hardware constraints on the sampling time, here $T_{\mathrm{s}} = 0.01$s.

Therefore LPV identification based on measured data can be used to identify a valid description of the system for which the efficient machinery of LPV control synthesis can be applied.

### *B. LPV-IO Identification/Optimization*

The intake manifold dynamics is affected by the pressure $p_1$, which is generated inside the manifold and the engine speed $p_2$. Therefore, both signals are considered as scheduling signals in the intended LPV model. Using data collected in the closed-loop setting of Fig. 2, different LPV-IO models are identified for the intake manifold of the SI engine with:

1) static-dependence on the scheduling signals: $\mathfrak{M}_{\mathrm{IO}}(p)$ as in (3).
2) dynamic-dependence on the scheduling signals: $\mathfrak{M}_{\mathrm{IO}}(q \diamond p)$ as in (26).
3) one-step backward dependence on the scheduling signals: $\mathfrak{M}_{\mathrm{IO}}(q^{-1}p)$ as in (31).
4) $n_a$-step backward dependence on the scheduling signals: $\mathfrak{M}_{\mathrm{IO}}(q^{-n_{\mathrm{a}}}p)$ as in (33).

All the coefficient functions are chosen as second-order polynomials in $p_1$ and $p_2$ with zero coefficients for the cross terms, e.g. in case of the model $\mathfrak{M}_{\mathrm{IO}}(p)$, which has static dependence, the coefficient functions are given for $i = 1 \dots, n_{\mathrm{a}}$, $j = 1, \dots, n_{\mathrm{b}}$ as

$$a_i(p) = a_{i0} + a_{i1}p_1 + a_{i2}p_1^2 + a_{i3}p_2 + a_{i4}p_2^2, \tag{56a}$$

$$b_j(p) = b_{j0} + b_{j1}p_1 + b_{j2}p_1^2 + b_{j3}p_2 + b_{j4}p_2^2. \tag{56b}$$

It is known that the intake manifold can be well represented by a second-order model, therefore $n_{\mathrm{a}}$ and $n_{\mathrm{b}}$ are both chosen to be 2. In the system identification stage, the constant coefficients $\{a_{il}\}_{i=1,l=0}^{2,4}$ and $\{b_{jl}\}_{j=1,l=0}^{2,4}$, see (56a-b), are estimated by a least-squares estimate. It is known that applying a direct estimate in closed loop via a least-squares approach results in a biased model. However, there exist many methods to overcome such an issue by applying instrumental variables, nonlinear optimization based techniques, etc. It should be noted that in the studied case here we do not take into account any noise source as our aim is to test whether the different model structures specified above are able to approximate the intake manifold dynamics. Therefore, the bias issue does not play a role in this setting. Hence identification is used as an optimization tool instead of estimation in a stochastic sense.

All LPV-IO model structures in (3), (26), (31) and (33) can be reformulated as

$$y(k) = \phi^\top(k)\theta, \tag{57}$$

where, for instance, with static dependence we have

$$\phi(k) = \varphi(k) \otimes P(k), \qquad \varphi(k) = \begin{bmatrix} -y(k-1) & -y(k-2) & u(k-1) & u(k-2) \end{bmatrix}^\top$$
$$P(k) = \begin{bmatrix} 1 & p_1(k) & p_1^2(k) & p_2(k) & p_2^2(k) \end{bmatrix}^\top$$

and the constant coefficients are collected in the vector $\theta \in \mathbb{R}^{20}$.

In order to gather informative signals for optimizing $\theta$, a white noise $N_{\text{ref}}$, see Fig. 2, is designed with uniform distribution $\mathcal{U}(760, 6250)$ and level change at random instances, which are specified by an additional random variable $\mathcal{U}(0, 1)$ for deciding when to change the level. Another signal $v$, see Fig. 2, with the same properties, but with $\mathcal{U}(10, 90)$, is designed to excite the input-output dynamics of the system (from $\alpha_{\text{lim}}$ to $m_{\text{nac}}$), see [35] for the probability characteristics of this class of signals. As a consequence, the pressure signal is internally generated. The required operating ranges for the different variables to design the input signal are as follows

$$y = m_{\text{nac}} \in [10, 90]\%, \qquad u = \alpha_{\text{lim}} \in [0, 100]\%,$$
$$p_1 \in [99, 950] \text{ hPa}, \qquad p_2 \in [760, 6250] \text{ rpm}.$$

The input-output data is collected and divided into estimation and validation sets.

In case of a measurement noise, (57) corresponds to the one-step ahead prediction of the output in a prediction-error identification setting under the assumption of an ARX noise model. Here we consider the noiseless case in which (57) gives the data equation of the model directly. Hence, for a set of input, scheduling signals and output data

$$Z^N := \big\{[u(k), p_1(k), p_2(k), y(k)],\ k = 1, \ldots, N\big\};$$

the least-squares parameter estimate for (57)

$$\hat{\theta}_N := \arg\min_{\theta \in \mathbb{R}^{20}} V_N(\theta, Z^N), \quad \text{where} \quad V_N(\theta, Z^N) := \frac{1}{N}\sum_{k=1}^{N}(y(k) - \phi^\top(k)\theta)^2, \tag{58}$$

can be obtained by linear regression.

The above given procedure has been adopted to calculate LPV-IO models for the intake manifold with different structures. Table I shows a comparison between the resulting models in terms of the BFR of the simulated outputs, where all of these models are simulated using the same set of validation data. It is clear that the obtained models with the structures (3), (26), (31) and (33) give almost the same BFR $\approx 95\%$ which shows that all these structures are approximating well the intake manifold plant.

TABLE I

BFR OF THE ESTIMATED MODELS FOR THE INTAKE MANIFOLD PLANT.

| | $\mathfrak{M}_{\mathrm{IO}}(p)$ | $\mathfrak{M}_{\mathrm{IO}}(q \diamond p)$ | $\mathfrak{M}_{\mathrm{IO}}(q^{-1}p)$ | $\mathfrak{M}_{\mathrm{IO}}(q^{-n_{\mathrm{a}}}p)$ |
|---|---|---|---|---|
| BFR % | 96.47150 | 95.72519 | 96.03974 | 93.95923 |

## *C. LPV-SS Realization*

Next, the identified LPV-IO models are transformed to LPV-SS forms by using the approaches introduced in Section IV. Regarding the model $\mathfrak{M}_{\mathrm{IO}}(q^{-1}p)$, a non-minimal 3rd-order LPV-SS model is produced via the augmented SS form. Therefore, the Ho-Kalman algorithm is employed to obtain a reduced 2rd-order LPV-SS model. It is worth to mention that the obtained non-minimal LPV-SS model is quadratically not stabilizable, which means that model order reduction based on balanced truncation with fixed similarity transformation like [23] can not be used to reduce the order of the model. In addition, without quadratic stabilizability and detectability properties, simple and practical LPV controllers based on quadratic $\mathcal{H}_\infty$ performance, e.g. [36], cannot be synthesized. Interestingly, the resulting reduced LPV-SS model based on the Ho-Kalman algorithm is balanced and quadratically stabilizable and detectable; this means that the lack of quadratic stabilizability of the non-minimal model is due to the 'additional' state, which has been removed by the Ho-Kalman algorithm. However the reduced LPV-SS model approximates the original model only with $\mathsf{BFR} = 75.58\%$.

Next, LPV controllers are synthesized based on all the transformed LPV-SS models, and the resulting controllers are implemented on the original nonlinear model of the intake manifold. This step is performed in order to assess the approximation quality of the different model structures and the different input-output to state-space transformations methods in terms of control design.

## *D. Control Design*

The linear matrix inequalities optimization-based LPV controller synthesis technique proposed in [36] is used to design controllers for the transformed LPV-SS models with a mixed-sensitivity loop shaping technique. This method is based on a polytopic representation of the LPV-SS models, therefore an affine representation of the system matrices as in (40a-b) is necessary. The performance requirements for the closed-loop are specified as:

- a rise time of $t_{\mathrm{r}} = 0.15$s, a settling time of $t_{\mathrm{s}} = 0.3$s.

TABLE II

BEST INDUCED $\ell_2$ GAIN FOR EACH DESIGN.

| | $K_{\mathfrak{M}_{\mathrm{IO}}(p)}$ | $K_{\mathfrak{M}_{\mathrm{IO}}(q\diamond p)}$ | $K_{\mathfrak{M}_{\mathrm{IO}}(q^{-1}p)}$ | $K_{\mathfrak{M}_{\mathrm{IO}}(q^{-n_{\mathrm{a}}}p)}$ |
|---|---|---|---|---|
| $\gamma$ | 20.3987 | 1.1205 | 1.1091 | 2.8549 |

- overshoot $M_{\mathrm{p}} < 5\%$, steady state error $e_\infty < 1\%$.
- constraint on actuator usage.

These objectives are translated into shaping filters

$$W_{\mathrm{S}} = \frac{0.02z + 0.02}{z - 0.9998}, \quad W_{\mathrm{KS}} = \frac{0.000643z - 0.0002439}{z + 0.9956},$$

to shape respectively the sensitivity and the control-sensitivity of the closed-loop. With these filters a generalized plant formulation of the models can be constructed. For consistent comparison, the same shaping filters are used for all synthesis problems. Then the mixed-sensitivity criterion is minimized such that the induced $\ell_2$ gain, of both the sensitivity and the control-sensitivity, is less than some prescribed value $\gamma > 0$. According to these considerations, controllers of order five have been computed and the best achieved performance index $\gamma$ for each design is given in Table II. The controller $K_{\mathfrak{M}_{\mathrm{IO}}(p)}$ is based on an LPV-SS model converted from the IO model $\mathfrak{M}_{\mathrm{IO}}(p)$ using the LTI rules. As a common property of the LMI-based synthesis approaches, the order of any of the designed controllers is equal to the order of the generalized plant that includes in addition to the system, the weighting filters and a first order pre-filter, which is used here since the $B$ matrix is parameter dependent, see [36] for more details.

### *E. Evaluation*

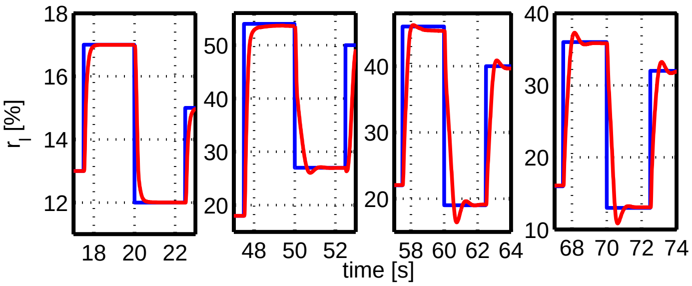
Finally, all controllers are applied to the nonlinear physical model of the intake manifold. Fig. 3a-e demonstrate the tracking of the normalized air charge $r_1 = m_{\mathrm{nac}}$ at different levels and periods of a specified typical trajectory with the different controllers $K_{\mathfrak{M}_{\mathrm{IO}}(p)}$, $K_{\mathfrak{M}_{\mathrm{IO}}(q\diamond p)}$, $K_{\mathfrak{M}_{\mathrm{IO}}(q^{-1}p)}$ and $K_{\mathfrak{M}_{\mathrm{IO}}(q^{-n_{\mathrm{a}}}p)}$, respectively. In general, with all designed controllers, except $K_{\mathfrak{M}_{\mathrm{IO}}(p)}$, $m_{\mathrm{nac}}$ follows the given reference trajectory in a satisfactory manner, with a rise time, maximum overshoot and steady state error within the limits which have been specified above. Even the controller $K_{\mathfrak{M}_{\mathrm{IO}}(q^{-1}p)}$, which has been designed based on the worst approximate model, namely the reduced augmented model, shows reasonable tracking with acceptable transient response, see Fig. 3c. The best performance is achieved by $K_{\mathfrak{M}_{\mathrm{IO}}(q\diamond p)}$. This is not surprising as the model $\mathfrak{M}_{\mathrm{IO}}(q \diamond p)$ have a high quality fit of $95\%$ and the used SS realization is exact, while in the $\mathfrak{M}_{\mathrm{IO}}(q^{-1}p)$ case the order reduction decreased the quality to $75\%$. The second best result is provided by $K_{\mathfrak{M}_{\mathrm{IO}}(q^{-n_\mathrm{a}}p)}$ due to the fact that the initial fit of $\mathfrak{M}_{\mathrm{IO}}(q^{-n_\mathrm{a}}p)$ was worse than $\mathfrak{M}_{\mathrm{IO}}(q \diamond p)$. On the other hand, the high-quality model $\mathfrak{M}_{\mathrm{IO}}(p)$ based controller $K_{\mathfrak{M}_{\mathrm{IO}}(p)}$, which has been synthesized based on a SS model constructed by the LTI rules, violates the constraints on $t_\mathrm{r}, t_\mathrm{s}, M_\mathrm{p}, e_\infty$, see Fig. 3a. In order to achieve the performance of the other controllers by $K_{\mathfrak{M}_{\mathrm{IO}}(p)}$, the shaping filters have been re-tuned and the best achieved performance is shown in Fig. 3e, which still violates the requirements on $M_\mathrm{p}$ in addition to the undesired oscillations in the control signal.

This example demonstrates the following: The proposed structures of LPV-IO models in (26), (31) and (33) can provide a good approximation of the original system and can be used for the purpose of control synthesis based on the associated realization approaches. Furthermore, using the LTI rules to obtain an SS realization for an LPV-IO model may lead to an inadequate approximation of the true dynamics, possibly resulting in a significant performance loss of the closed-loop control.

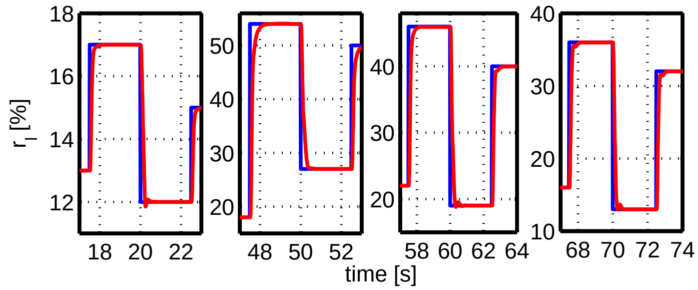
Based on the previous discussion the following practical advices can be given: In order to ensure low complexity of the control design phase, LTI conversion rules to get the SS realization should be tested first by adopting the criterion in Section III. If it turns out that the LTI rules provide a poor SS realization, an LPV-IO model with any of the structures in Subsection IV-A or IV-C can be identified and via the proposed SS realization approaches an adequate LPV-SS model of the plant can be obtained with static dependence. As these realizations are exact, initial model fit indicates the expected performance of the designed controller on the true system. If these structures do not result in an acceptable model for the underlying process, then the structure in Subsection IV-B can be adopted with the model reduction approach in Section V to provide minimal SS realization on the possible expanse of model quality.

## VII. Conclusions

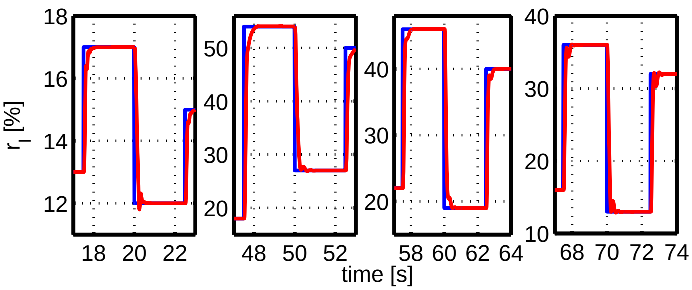
An equivalent representation of an LPV-IO model in LPV-SS form, in general necessitates dynamic dependence of the state-space realization on the scheduling signal. Neglecting this fact can cause a significant performance loss in the control design phase. On the other hand,



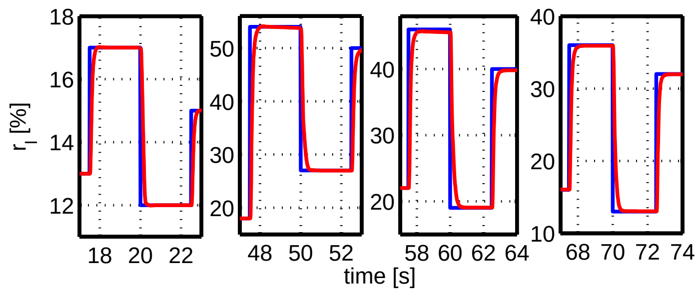

(a) $K_{\mathfrak{M}_{\mathrm{IO}(p)}}$



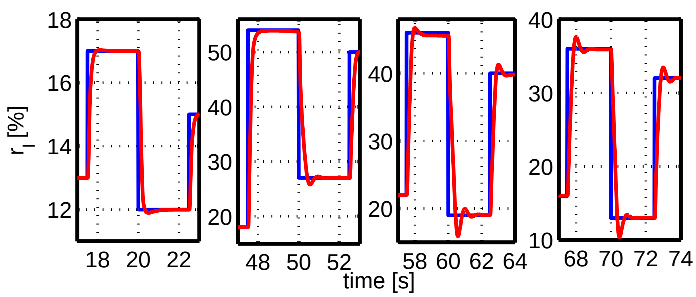

(b) $K_{\mathfrak{M}_{\mathrm{IO}(q\diamond p)}}$




(c) $K_{\mathfrak{M}_{\mathrm{IO}(q^{-1}p)}}$




(d) $K_{\mathfrak{M}_{\mathrm{IO}(q^{-n_{\mathrm{a}}}p)}}$




(e) Improved controller $K_{\mathfrak{M}_{\mathrm{IO}(p)}}$

Fig. 3. Tracking of $m_{\mathrm{nac}}$ with the designed controllers.

the presence of dynamic dependence leads to difficulties in terms of controller design and implementation. To deal with this problem, practical and systematic methods have been proposed. First a criterion has been introduced to decide when the LTI conversion rules can be used without serious consequences and how much can be lost in terms of model validity. Then a realization algorithm was provided that is capable of yielding a minimal state-space realization of any LPV-IO model with the introduction of dynamic dependence. In order to avoid dynamic dependence of the resulting SS realization, four pragmatic approaches have been proposed to convert LPV-IO models to LPV-SS realizations without introducing dynamic dependence. It was observed that in some situations the restriction of static dependence in the realization results in a non-minimal SS model. To obtain an economical SS representation of such augmented SS models even if they are unstable or not quadratically stabilizable and detectable, an LPV Ho-Kalman type of model reduction approach has been developed. Finally applicability and usefulness of all ideas have been demonstrated by solving the charge control problem of an SI engine.

## References

[1] A. Marcos and G. J. Balas, "Development of linear-parameter-varying models for aircraft," *Journal of Guidance, Control and Dynamics*, vol. 27, no. 2, pp. 218–228, 2004.

[2] G. Belforte, F. Dabbene, and P. Gay, "LPV approximation of distributed parameter systems in environmental modeling," *Environmental Modeling & Software*, vol. 20, no. 8, pp. 1063–1070, 2005.

[3] P. Apkarian and P. Gahinet, "A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers," *IEEE Trans. on Automatic Control*, vol. 40, no. 5, pp. 853–864, 1995.

[4] C. W. Scherer, "Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control for time-varying and linear parametrically-varying systems," *Int. Journal of Robust and Nonlinear Control*, vol. 6, no. 9-10, pp. 929–952, 1996.

[5] F. Wu and K. Dong, "Gain-scheduling control of LFT systems using parameter-dependent Lyapunov functions," *Automatica*, vol. 42, no. 1, pp. 39–50, 2006.

[6] H. Abbas, S. Chughtai, and H. Werner, "A hybrid gradient-LMI algorithm for the synthesis of LPV gain-scheduled controllers," in *Proc. of the European Control Conf.*, Budapest, Hungary, Aug. 2009, pp. 3407–3412.

[7] R. Tóth, *Modeling and Identification of Linear Parameter-Varying Systems*, ser. Lecture Notes in Control and Information Sciences, Vol. 403. Springer-Germany, 2010.

[8] B. Bamieh and L. Giarré, "Identification of linear parameter varying models," *Int. Journal of Robust and Nonlinear Control*, vol. 12, pp. 841–853, 2002.

[9] 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, July 2008, pp. 4018–4023.

[10] H. Abbas and H. Werner, "An instrumental variable technique for open-loop and closed-loop identification of input-output LPV models," in *Proc. of the European Control Conf.*, Budapest, Hungary, 2009, pp. 2646–2651.

[11] V. Laurain, M. Gilson, R. Tóth, and H. Garnier, "Refined instrumental variable methods for identification of LPV Box-Jenkins models," *Automatica*, vol. 46, no. 6, pp. 959–967, 2010.

[12] M. G. Wassink, M. van de Wal, C. W. Scherer, and O. Bosgra, "LPV control for a wafer stage: Beyond the theoretical solution," *Control Engineering Practice*, vol. 13, no. 2, pp. 231–245, 2004.

[13] L. Giarré, D. Bauso, P. Falugi, and B. Bamieh, "LPV model identification for gain scheduling control: An application to rotating stall and surge control problem," *Control Engineering Practice*, vol. 14, no. 4, pp. 351–361, 2006.

[14] Y. Bolea, J. Blesa, and V. Puig, "LPV modeling and identification of an open canal for control," in *Proc. of the European Control Conf.*, Kos, Greece, Aug. 2007, pp. 3584–3591.

[15] A. Kwiatkowski, H. Werner, and M. T. Boll, "Automated generation and assessment of affine LPV models," in *Proc. of the 45th IEEE Conf. on Decision and Control*, San Diego, California, USA, Dec. 2006, pp. 6690–6695.

[16] R. Tóth, F. Felici, P. S. C. Heuberger, and P. M. J. Van den Hof, "Discrete time LPV I/O and state space representations, differences of behavior and pitfalls of interpolation," in *Proc. of the European Control Conf.*, Kos, Greece, July 2007, pp. 5418–5425.

[17] R. Tóth, J. C. Willems, P. S. C. Heuberger, and P. M. J. Van den Hof, "The behavioral approach to linear parameter-varying systems," *in print, IEEE Trans. on Automatic Control*, 2011.

[18] J. C. Willems, "The behavioral approach to open and interconnected systems," *IEEE Control Systems Magazine*, vol. 27, no. 6, pp. 46–99, 2007.

[19] L. Ljung, *System Identification, theory for the user*. Prentice Hall, 1999.

[20] ——, *System Identification Toolbox, for use with Matlab*. The Mathworks Inc., 2006.

[21] D. R. and S. Khatri, "Kalman decomposition of linear fractional transformation representations and minimality," in *Proc. of the American Control Conf.*, Albulquerque, New Mexico, USA, June 1997, pp. 3557–3561.

[22] C. Beck, "Coprime factors reduction methods for linear parameter varying and uncertain systems," *Systems & Control Letters*, vol. 55, no. 3, pp. 199–213, 2006.

[23] G. D. Wood, P. J. Goddard, and I. Glover, "Approximation of linear parameter-varying systems," in *Proc. of the 35th IEEE Conf. on Decision and Control*, Kobe, Japan, Dec. 1996, pp. 406–411.

[24] S. Hecker, A. Varga, and J. Magni, "Enhanced LFR-toolbox for Matlab," *Aerospace Science and Technology*, vol. 9, no. 2, pp. 173–180, 2005.

[25] J. F. Magni, "User manual of the linear fractional representation toolbox version 2.0," Systems Control and Flight Dynamics Department, Onera, Tech. Rep. TR 5/10403.01F DCSD, 2006.

[26] S. Gugercin and A. C. Antoulas, "A comparative study of 7 algorithms for model reduction," in *Proc. of the 39th IEEE Conf. on Decision and Control*, Sydney, Australia, Dec. 2000, pp. 2367–2372.

[27] A. Varga, "Coprime factors model reduction based on accuracy enhancing techniques," *Systems Analysis Modeling and Simulation*, vol. 11, pp. 303–311, 1993.

[28] ——, "Enhanced modal approach for model reduction," *Mathematical Modeling of Systems*, vol. 1, pp. 91–105, 1995.

[29] B. de Schutter, "Minimal state-space realization in linear system theory: an overview," *Journal of Computational and Applied Mathematics*, vol. 121, no. 1-2, pp. 331–354, 2000.

[30] F. Felici, J. W. van Wingerden, and M. Verhaegen, "Subspace identification of MIMO LPV systems using a periodic scheduling sequence," *Automatica*, vol. 43, no. 10, pp. 1684–1697, 2006.

[31] 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.

[32] L. M. Silverman and H. E. Meadows, "Controllability and observability in time-variable linear systems," *SIAM Journal on Control*, vol. 5, no. 1, pp. 64–73, 1967.

[33] S. Hecker, A. Varga, and J. Magni, "Krylov space methods on state-space control models," *Circuits, Systems and Signal Processing*, vol. 13, no. 6, pp. 733–758, 1994.

[34] A. Kwiatkowski, "LPV modeling and application of LPV controllers to SI engines," Ph.D. dissertation, Hamburg University of Technology, Germany, 2008.

[35] M. Norgaard, O. Ravn, N. Poulsen, and L. Hansen, *Neural Networks for Modeling and Control of Dynamic Systems*. Springer-Verlag, 2000.

[36] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.