

A behavioral approach for LPV data-driven representations

Chris Verhoek, Ivan Markovsky, Sofie Haesaert, and Roland Tóth

Abstract—In this paper, we present a data-driven representation for linear parameter-varying (LPV) systems, which can be used for direct data-driven analysis and control of such systems. Specifically, we use the behavioral approach to develop a data-driven representation of the finite-horizon behavior of LPV systems for which there exists a kernel representation with *shifted-affine* scheduling dependence. Moreover, we provide a necessary and sufficient rank-based test on the available data that concludes whether the data *fully* represents the finite-horizon LPV behavior. Using the proposed data-driven representation, we also solve the data-driven simulation problem for LPV systems. Through multiple examples, we demonstrate that the results in this paper allow us to formulate a novel set of direct data-driven analysis and control methods for LPV systems, which are also applicable for LPV embeddings of nonlinear systems.

Index Terms—Behavioral systems theory, linear parameter-varying systems, data-driven simulation and control.

I. INTRODUCTION

Due to the ever-growing complexity of engineering systems, hindering traditional modeling methods, and the increasing availability of data there is a growing interest in accomplishing analysis and control design directly on the basis of data. Particularly, direct data-driven analysis and control methods founded on the behavioral framework have gained a lot of attention. This is because these methods allow for rigorous stability and performance guarantees. A key result that enables using such approaches is Willems' Fundamental Lemma [1], which allows for representing the behavior of a *discrete-time* (DT) *linear time-invariant* (LTI) system using a single sequence of measurement data, where the input is *persistently exciting* (PE) of a certain order, i.e., the data is sufficiently ‘rich’. Based on this result, many data-driven analysis and control methods have been developed for DT LTI systems, see,

This work has been supported by The MathWorks Inc. and by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002). Opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the MathWorks Inc. or the European Union.

C. Verhoek, R. Tóth and S. Haesaert are with the Control Systems Group, Eindhoven University of Technology, The Netherlands. I. Markovsky is with the Catalan Institution for Research and Advanced Studies (ICREA), Barcelona, and the International Centre for Numerical Methods in Engineering (CIMNE), Barcelona, Spain. R. Tóth is also with the HUN-REN Institute for Computer Science and Control, Hungary. Email addresses: {c.verhoek, r.toth, s.haesaeert}@tue.nl, and imarkovsky@cimne.upc.edu. Corresponding author: Chris Verhoek.

e.g., [2] for an overview. A further generalization of the PE condition is the so-called *generalized persistence of excitation* (GPE) condition [3], which provides an (a posteriori) test to decide whether the *data* is rich enough to represent the underlying behavior of the system. In the literature, ‘the Fundamental Lemma’ is generally used to refer to the data-driven representation of the finite-horizon behavior, irrespective of whether it is used in conjunction with the PE condition or the GPE condition.

Many extensions of the Fundamental Lemma have been proposed in the literature, such as for *continuous-time* (CT) LTI systems [4], 2D LTI systems [5], stochastic LTI systems [6], and variants for convex, conical, and affine behaviors [7]. Beyond the class of LTI systems, several approaches have been proposed to extend the Fundamental Lemma towards specific classes of nonlinear systems, e.g., [8]–[11]. These methods, however, generally impose restrictive assumptions on the system class (e.g., the system must be feedback linearizable, periodic, or described by a finite Volterra series). These assumptions allow for recasting the problem in an equivalent LTI form on which the standard PE or GPE conditions can be applied.

Alternatively, an efficient way of handling nonlinear systems is via the framework of *linear parameter-varying* (LPV) systems. For LPV systems, the behavior is defined by a linear dynamic relationship that depends on a *measurable* signal p , called the *scheduling signal* [12], [13]. Such LPV representations are often used as linear, surrogate models for the analysis and control of nonlinear systems, where the scheduling signal captures the nonlinearities and exogenous effects [14], [15]. This makes the LPV framework a bridge between linear and nonlinear analysis and control [13]. In fact, the LPV framework has been successfully applied in practice to solve complex nonlinear control problems [16]–[18]. We refer to [12] for a broad, up-to-date overview of the current state-of-the-art in (model-based) LPV analysis and control. Due to the attractive properties and success of the LPV framework, developing direct data-driven analysis and control methods for LPV systems is an important stepping stone to achieve a generalization of the original LTI data-driven results to the nonlinear case. In this paper, we aim to accomplish this in terms of generalization of the behavioral data-driven representation for the class of LPV systems and developing a corresponding GPE condition.

To formulate our results, we will restrict the scope to LPV systems whose behavior can be characterized by an LPV kernel representation that has *shifted-affine* scheduling

dependence. We refer to this subclass as LPV-SA systems. This is a highly useful subclass, as these systems admit a direct LPV *state-space* (SS) representation with static scheduling dependence, which is convenient for usage in LPV analysis and control methods. Moreover, through the 2nd Fundamental Theorem of Calculus, a large class of nonlinear systems¹ can be modeled in an LPV-SA form [19], [20].

In [21], a rather general extension of the Fundamental Lemma was given for the class of LPV systems with meromorphic, dynamic scheduling dependence. However, beyond the theoretical interest, this formulation results in a data-driven representation that uses composition weights for the data that are arbitrary meromorphic functions of the scheduling signal with an arbitrary number of lags. As there are no systematic methods available that can compute these meromorphic weights, the data-driven LPV representation in [21] is difficult to apply in practice. It was also shown as a remark in [21], that this general form of the LPV Fundamental Lemma can be reduced to a simpler, practically useful form for LPV-SA systems, based on which a whole series of contributions on data-driven LPV methods has been developed. Particularly, direct data-driven predictive control schemes [22], [23], direct data-driven LPV state-feedback control synthesis [24], direct data-driven dissipativity analysis [25], or efficient identification schemes [26]. However, the LPV Fundamental Lemma for this rather useful subclass of LPV-SA systems has never been directly formulated. Furthermore, *computable* conditions to check whether the data is ‘rich’ enough, i.e., the derivations of GPE conditions, have never been sorted out. Moreover, the LPV data-driven simulation problem has not been formally solved yet. In this paper, we fill these gaps in the current literature by providing the following contributions:

- C1: We formulate a finite-horizon data-driven LPV representation for LPV-SA systems that is directly and easily computable from a given data set;
- C2: We provide a necessary and sufficient condition that is verifiable on the given data set to conclude whether the data can characterize the full finite-horizon behavior of the LPV-SA system;
- C3: We provide a formal solution to the data-driven simulation problem;
- C4: We demonstrate the capabilities of our results in terms of solving simulation and control problems on an LPV system and a nonlinear system.

The remainder of the paper is structured as follows. We formalize the considered problem setting in Section II. Section III discusses the properties of behaviors of LPV-SA systems, such as their complexity and dimension. The data-driven representation and associated conditions on the data, constituting to Contributions C1 and C2, are given in Section IV. The formalization of the simulation problem and its solution (Contribution C3) are given in Section V, while Section VI presents two examples that demonstrate the applicability of our methods in analysis and control of LPV and nonlinear systems.

¹Specifically, the class of nonlinear DT systems with a continuously differentiable input-output map. That is, for $y(k) = f(y(k-1), \dots, y(k-n), u(k), \dots, u(k-n))$, with $f \in \mathcal{C}_1$.

Finally, conclusions and possible future research directions are given in Section VII.

Notation:

\mathbb{R} is the set of real numbers, while the set of integers is given by \mathbb{Z} . Consider the subspaces \mathbb{A}, \mathbb{B} . The projection of $\mathbb{D} \subseteq \mathbb{A} \times \mathbb{B}$ onto the elements of \mathbb{A} is denoted by $\pi_{\mathbb{A}}\mathbb{D} = \{a \in \mathbb{A} \mid (a, b) \in \mathbb{D}\}$, while $\mathbb{B}^{\mathbb{A}}$ indicates the collection of all maps from \mathbb{A} to \mathbb{B} . We denote the dimension of a subspace by $\dim(\mathbb{A})$. The p -norm of a vector $x \in \mathbb{R}^{n_x}$ is denoted by $\|x\|_p$. For the two matrices $A \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{p \times q}$, the Kronecker product is given as $A \otimes B \in \mathbb{R}^{pm \times qn}$, while blkdiag is the block-diagonal composition for matrices, i.e., $\text{blkdiag}(A, B) = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} \in \mathbb{R}^{n+p \times m+q}$. The identity matrix of size $n \times n$ is denoted as I_n . Furthermore, $\text{col}(x_1, \dots, x_n)$ denotes the column vector $[x_1^\top \dots x_n^\top]^\top$. The image (or column space) of a matrix A is denoted as $\text{image}(A)$, while its kernel (or null space) is denoted as $\text{kernel}(A)$. Consider a signal $w : \mathbb{Z} \rightarrow \mathbb{R}^{n_w}$. The value of a signal $w : \mathbb{Z} \rightarrow \mathbb{R}^{n_w}$ at time step k is denoted as $w(k) \in \mathbb{R}^{n_w}$ and its i^{th} element is given by $w_i(k) \in \mathbb{R}$. The forward and backward time-shift operators are denoted by q and q^{-1} . We denote a time-interval between t_1 and t_2 , $t_1 \leq t_2$ by $[t_1, t_2] \subset \mathbb{Z}$. For the time interval $\mathbb{T} \subseteq \mathbb{Z}$, we write $w_{\mathbb{T}}$ as the truncation to w on \mathbb{T} , e.g., for $\mathbb{T} := [1, N]$ we have $w_{[1, N]} = (w(1), \dots, w(N)) \in (\mathbb{R}^{n_w})^{[1, N]}$. The notation $w_{[1, N_1]} \wedge v_{[1, N_2]} \in (\mathbb{R}^{n_w})^{[1, N_1+N_2]}$ indicates concatenation of $w_{[1, N_1]}$ followed by $v_{[1, N_2]} \in (\mathbb{R}^{n_w})^{[1, N_2]}$, while, with a slight abuse of notation, $\text{col}(w, v)$ indicates the stacked signal $(\dots, [w(k-1)], [w(k)], [w(k+1)], \dots, [v(k-1)], [v(k)], [v(k+1)], \dots)$. A sequence of the following form $(p(k) \otimes w(k))_{k=1}^N$ is denoted by $w_{[1, N]}^p$. For $w_{[1, N]}$, the associated Hankel matrix of depth L is given by

$$\mathcal{H}_L(w_{[1, N]}) = \begin{bmatrix} w(1) & w(2) & \dots & w(N-L+1) \\ w(2) & w(3) & \dots & w(N-L+2) \\ \vdots & \vdots & \ddots & \vdots \\ w(L) & w(L+1) & \dots & w(N) \end{bmatrix},$$

while the block-diagonal Kronecker operator ‘ \odot ’ is denoted as $w_{[1, N]} \odot I_n = \text{blkdiag}(w(1) \otimes I_n, \dots, w(N) \otimes I_n)$. Finally, throughout the paper we distinguish signals from recorded data sets with a breve accent, e.g., \check{w} .

II. PROBLEM FORMULATION

A. System definition and behaviors

We study DT LPV systems that can be represented by the *kernel* representation:

$$\underbrace{\sum_{i=0}^{n_r} r_i(q^i p) q^i}_{R(p, q)} w = 0, \quad (1a)$$

with manifest signals $w \in (\mathbb{R}^{n_w})^{\mathbb{Z}}$, scheduling signals $p \in \mathbb{P}^{\mathbb{Z}}$ and scheduling dependent coefficients $r_i : \mathbb{P}^{\mathbb{Z}} \rightarrow \mathbb{R}^{n_k \times n_w}$ that have a *shifted-affine* dependence on p :

$$r_i(q^i p) = r_{i,0} + \sum_{j=1}^{n_p} r_{i,j} q^i p_j, \quad (1b)$$

with $r_{i,j} \in \mathbb{R}^{n_k \times n_w}$. We refer to LPV systems that have a representation in the form of (1) as *LPV Shifted-Affine* (LPV-SA) systems. The signal p is considered to be free and varying in the *scheduling set* $\mathbb{P} \subseteq \mathbb{R}^{n_p}$, which is often chosen as a closed subset of \mathbb{R}^{n_p} that contains the origin. By introducing a partitioning² of w to inputs $u \in (\mathbb{R}^{n_u})^{\mathbb{Z}}$ (maximally free³ signals) and outputs $y \in (\mathbb{R}^{n_y})^{\mathbb{Z}}$, (1b) becomes an *input-output* (IO) representation with $w = \text{col}(u, y)$ and $r_i = [r_{u,i} \ r_{y,i}]$, where $r_{u,i} : \mathbb{P}^{\mathbb{Z}} \rightarrow \mathbb{R}^{n_k \times n_u}$, $r_{y,i} : \mathbb{P}^{\mathbb{Z}} \rightarrow \mathbb{R}^{n_k \times n_y}$ and $n_w = n_u + n_y$. For the remainder of this paper, the class of LPV-SA systems with n_p scheduling signals and n_w manifest variables is denoted by Σ_{n_p, n_w} .

In this paper, we consider the *behavioral approach* [27]. The behavior \mathfrak{B} is the collection of all solution trajectories compatible with the system. The representation (1) is a representation of a given behavior \mathfrak{B} of an LPV-SA system $\Sigma \in \Sigma_{n_p, n_w}$ if

$$\mathfrak{B} = \{(w, p) \in (\mathbb{R}^{n_w} \times \mathbb{P})^{\mathbb{Z}} \mid (1) \text{ holds}\}. \quad (2)$$

Following [14, Def. 3.24 & Def. 3.25], we consider (1) to be minimal if n_k is such that $R(p, q)$ in (1) has full row rank, and this full row rank representation has the smallest polynomial order n_r among all possible full row rank kernel representations that can characterize \mathfrak{B} of a $\Sigma \in \Sigma_{n_p, n_w}$. It is shown in [14, Thm. 3.6] that there always exists a minimal kernel representations that represents \mathfrak{B} .

A few subsets of \mathfrak{B} are useful to consider. Specifically, the set of admissible scheduling trajectories of \mathfrak{B} :

$$\mathfrak{B}_{\mathbb{P}} = \pi_p \mathfrak{B} = \{p \in \mathbb{P}^{\mathbb{Z}} \mid \exists w \in (\mathbb{R}^{n_w})^{\mathbb{T}} \text{ s.t. } (w, p) \in \mathfrak{B}\}, \quad (3)$$

the set of w trajectories that are compatible with a particular, given scheduling trajectory $p \in \mathfrak{B}_{\mathbb{P}}$, i.e.,

$$\mathfrak{B}_p = \{w \in (\mathbb{R}^{n_w})^{\mathbb{T}} \mid (w, p) \in \mathfrak{B}\}, \quad (4)$$

and the trajectories in \mathfrak{B} that are restricted to the time interval $[k_1, k_2] \subset \mathbb{Z}$, $k_1 \leq k_2$. The set containing these trajectories is given by

$$\begin{aligned} \mathfrak{B}|_{[k_1, k_2]} = \{&(w, p)_{[k_1, k_2]} \in (\mathbb{R}^{n_w} \times \mathbb{P})^{[k_1, k_2]} \mid \\ &\exists (\omega, \rho) \in \mathfrak{B} \text{ s.t. } (w(k), p(k)) = (\omega(k), \rho(k)) \\ &\text{for } k_1 \leq k \leq k_2\}. \end{aligned}$$

Note that this notation can be applied to (3) and (4) as well.

Because LPV systems are linear along a scheduling trajectory, \mathfrak{B}_p is a linear subspace. LPV systems are time invariant in the sense that $q\mathfrak{B} = \mathfrak{B}$, and thus $q\mathfrak{B}_p = \mathfrak{B}_{qp}$. Moreover, $\mathfrak{B}|_{[k_1, k_2]} = \mathfrak{B}|_{[1, k_2 - k_1 + 1]}$.

Example 1. We will clarify the introduced concepts and notation by means of an example. We consider a *mass-spring-damper* (MSD) system where the stiffness of the spring is varying with a measurable signal p . Such an MSD system can occur in many practical scenarios, e.g., the linear axis of a

²In many works on the behavioral approach, a non-singular permutation matrix Π is used to characterize the partitioning, such that $w = \Pi \begin{bmatrix} u \\ y \end{bmatrix}$. To streamline the notation, we choose w.l.o.g. $\Pi = I$.

³A maximally free input means that for a given u , none of the components of y can be chosen freely for all $p \in \mathfrak{B}_{\mathbb{P}}$.

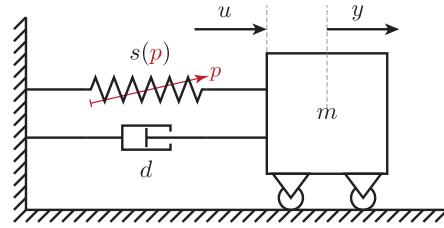


Fig. 1: Schematics of a mass-spring-damper system with a spring that is varying with a measurable signal p . The input of the system is a force u , while the output is the measured position y .

water-jet cutter where the stiffness of the cable-slab is dependent on the water-flow of the jet, hence p is the (measurable) water-flow. A schematic representation of the MSD system is given in Fig. 1. The input u is the force [N] exerted on the mass m [kg], and the output y is the measured position [m] of the mass. The damper has damping coefficient d , while the spring has a stiffness coefficient described by the function $s(p)$. The variation of $s(p)$ is described by

$$s(p(k)) = s_0 + s_1 p(k), \quad p(k) \in [-1, 1].$$

Euler discretization of the CT MSD dynamics with sampling time T_s gives the following LPV representation:

$$\begin{aligned} y(k) + \left(\frac{d T_s}{m} - 2 \right) y(k-1) \\ + \left(1 + \frac{s_0 T_s^2 - d T_s}{m} + \frac{s_1 T_s^2}{m} p(k-2) \right) y(k-2) \\ = \frac{T_s^2}{m} u(k-2). \quad (5) \end{aligned}$$

With $w = \text{col}(u, y)$, we can write (5) as an LPV-SA kernel representation of the form (1), with

$$R(p, q) = \left[-\frac{T_s^2}{m} \quad q^2 + \frac{d T_s - 2m}{m} q + \frac{m + s_0 T_s^2 - d T_s}{m} + \frac{s_1 T_s^2}{m} p \right],$$

where we first multiplied both sides in (5) with q^2 . The solution trajectories for this system, collected in \mathfrak{B} are described by this particular $R(p, q)$. \blacktriangleleft

B. Problem statement

The goal of this paper is to characterize $\mathfrak{B}|_{[1, L]}$ purely based on a given data set (Contribution C1). It is important that the characterization is *computable* and it is verifiable whether the full $\mathfrak{B}|_{[1, L]}$ is represented by the given data (Contribution C2). Once we have these, we can use the representation for data-driven simulation (Contribution C3). This gives the following problem formulations:

Problem 1. Given a data set from an LPV-SA system $\Sigma \in \Sigma_{n_p, n_w}$ with behavior \mathfrak{B} and a given complexity:

$$\mathcal{D}_{N_d} = (\check{w}_{[1, N_d]}, \check{p}_{[1, N_d]}) \in \mathfrak{B}|_{[1, N_d]}, \quad (6)$$

where \check{w} and \check{p} are noise free. For a given L , formulate a representation of $\mathfrak{B}|_{[1, L]}$ of Σ from only the data in \mathcal{D}_{N_d} .

To ensure validity of the representation, we require computationally verifiable conditions to conclude whether \mathcal{D}_{N_d} is ‘rich’ enough to represent $\mathfrak{B}|_{[1, L]}$:

Problem 2. Give an explicit condition for \mathcal{D}_{N_d} that is computationally verifiable to conclude whether $\mathfrak{B}|_{[1,L]}$ of $\tilde{\Sigma}$ can be fully represented by \mathcal{D}_{N_d} .

As motivated in Section I, the data-driven simulation problem for LPV systems was not formally solved. In this paper, we fill this gap by solving the following problem:

Problem 3. Use the solutions to Problems 1 and 2 to achieve simulation of $\tilde{\Sigma}$, i.e., computation of its response for a given input trajectory and initial condition, purely based on \mathcal{D}_{N_d} .

To obtain the solutions to the above problems, we first need to study the properties of LPV-SA systems and their (restricted) behaviors.

III. PROPERTIES OF LPV-SA BEHAVIORS

In this section, we explore the properties of LPV-SA behaviors that are instrumental to solve Problems 1–3. We first provide the connections between kernel, IO and SS representations of LPV-SA behaviors. These connections naturally lead to the formal notion of complexity for an LPV-SA system, with which we formulate one of the key ingredients required to solve Problem 2; the dimension of $\mathfrak{B}_p|_{[1,L]}$.

A. Representations of LPV-SA behaviors

As motivated in Section I, the class of LPV-SA systems is, although restrictive, a highly useful system class. This is because it has direct IO and SS realizations with a structured dependence, which streamlines LPV identification, analysis and control design.

1) *Input-output representations*: A key aspect of the behavioral framework is that there is no prior distinction between inputs and outputs. This makes that kernel representations, e.g., (1), are the fundamental building blocks for representing systems in this framework. In control engineering, however, defining the input and output properties of signals is often needed. Hence, following the partitioning $w = \text{col}(u, y)$, as already introduced in Section II, we can represent the behavior in terms of an LPV-IO form.

If the input is chosen such that it is maximally free, then the number of inputs n_u is an invariant property of \mathfrak{B} , which follows directly from [28, Thm. 6] and the definition of a maximally free input. We denote the invariant property of the input dimension by $\mathbf{m}(\mathfrak{B})$, and this is, in fact, the first component of the *complexity* of \mathfrak{B} .

We will consider an LPV-IO representation of the behavior in *filter form*, resulting from the transformation $q^{-n_r} R(p, q) w = 0$. In this filter form, we can split up the coefficients of (1) according to the aforementioned partitioning, providing the p -dependent coefficient functions

$$r_i(q^i p) = [b_{n_r-i}(q^{i-n_r} p) \quad -a_{n_r-i}(q^{i-n_r} p)], \quad i \in \mathbb{I}_0^{n_r},$$

with

$$a(p, q) = \sum_{i=0}^{n_a} a_i(q^{-i} p) q^{-i}, \quad b(p, q) = \sum_{i=0}^{n_b} b_i(q^{-i} p) q^{-i}, \quad (7)$$

and $n_r = \max(n_a, n_b)$, giving the shifted-affine LPV-IO realization

$$y(k) + \sum_{i=1}^{n_a} a_i(p(k-i)) y(k-i) = \sum_{j=0}^{n_b} b_j(p(k-j)) u(k-j). \quad (8)$$

Here, we assume that $a_0 = I_{n_y}$. Although for *single-input-single-output* (SISO) systems, dividing by a_0 is always possible to obtain (8), such an operation can potentially introduce rational dependency over $p(k)$ in the remaining coefficients with corresponding singularities. In the *multiple-input-multiple-output* (MIMO) case, invertibility of a_0 is not always ensured to obtain such a form. Nevertheless, in the sequel, we restrict our attention to cases where the leading coefficient a_0 can be taken to be identity, for the purpose of showing the connections between the different representations.

With $a_0 = I_{n_y}$, the scheduling-dependent functions a_i, b_i in (8) are simply partitions of the coefficient functions r_i in (1), hence they are affine in $p(k-i)$, and u is a maximally free signal w.r.t. each \mathfrak{B}_p . Minimality of the LPV-IO representation (8) is directly adopted from the kernel representation, and is achieved if a in (7) has full row rank. The LPV-IO representation is controllable if it is minimal and the polynomials a and b are *left-coprime*, see Def. 3.28 and the proof of Thm. 5.1 in [14]. The latter implies that a and b contain the minimum number of lags of u and y to represent \mathfrak{B} . Here we uncover another invariant property of \mathfrak{B} ; the *lag* $\mathbf{L}(\mathfrak{B})$, which is the minimum required degree of the polynomials in (7) to be able to represent \mathfrak{B} . Hence, for a minimal (8), $\max(n_a, n_b) = n_r = \mathbf{L}(\mathfrak{B})$. The lag is another component of the complexity of \mathfrak{B} . The final measure of the complexity of \mathfrak{B} is the minimal required state dimension of an LPV-SS realization of \mathfrak{B} , which we will discuss next.

2) *State-space representations*: LPV-SS representations are standard in LPV analysis and control design. Particularly useful are LPV-SS representations with *static* scheduling dependence, i.e., composed of matrices that are only dependent on $p(k)$:

$$qx = A(p)x + B(p)u, \quad (9a)$$

$$y = C(p)x + D(p)u, \quad (9b)$$

with $x(k) \in \mathbb{R}^{n_x}$ being a latent variable that qualifies as a state and $u(k) \in \mathbb{R}^{\mathbf{m}(\mathfrak{B})}$. The *full behavior* of (9) is given by

$$\mathfrak{B}^{\text{ss}} = \{(u, y, p, x) \in (\mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \times \mathbb{P} \times \mathbb{R}^{n_x})^{\mathbb{Z}} \mid (9) \text{ holds}\}. \quad (10)$$

An important feature of LPV-SA systems is that the LPV-IO representation (8) has a *direct* LPV-SS realization of the form (9) [29] with matrix functions

$$\begin{bmatrix} A(p) & B(p) \\ C(p) & D(p) \end{bmatrix} = \begin{bmatrix} -a_1(p) & I_{n_y} & \cdots & 0 & b_1(p) - a_1(p)b_0(p) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -a_{n_a-1}(p) & 0 & \cdots & I_{n_y} & b_{n_b-1}(p) - a_{n_b-1}(p)b_0(p) \\ -a_{n_a}(p) & 0 & \cdots & 0 & b_{n_b}(p) - a_{n_b}(p)b_0(p) \\ \hline I_{n_y} & 0 & \cdots & 0 & b_0(p) \end{bmatrix}, \quad (11)$$

where, with a slight abuse of notation, the state construction is, see [29, Sec. IV.A],

$$x_i = qx_{i-1} + a_{i-1}(p)y - b_{i-1}(p)u, \quad x_i(k) \in \mathbb{R}^{n_y} \quad (12a)$$

for $i \in [2, \max(n_a, n_b)]$, and where

$$x_1 = y - b_0(p)u. \quad (12b)$$

Note that with the construction (12), $n_x = n_y \mathbf{L}(\mathfrak{B})$.

Due to the direct LPV-SS realization, we have that $\pi_{u,y,p} \mathfrak{B}^{\text{ss}} = \mathfrak{B}$, i.e., the manifest behaviors defined by the IO representation (8) and the LPV-SS representation (9) with (11) are equivalent. Based on this, we can also define minimality of the LPV-SS representation (9) (with static scheduling dependence) as the minimum number of states required such that $\pi_{u,y,p} \mathfrak{B}^{\text{ss}} = \mathfrak{B}$ holds within the class Σ_{n_p, n_w} . We call this the *order* of \mathfrak{B} . This is an invariant property for behaviors \mathfrak{B} of LPV-SA systems in Σ_{n_p, n_w} , which follows from [14, Thm. 3.7]. This, in fact, is the last measure for the complexity of an LPV-SA system, which we denote by $\mathbf{n}(\mathfrak{B})$.

For SISO systems, i.e., $n_u = n_y = 1$, the direct LPV-SS realization (11) is minimal, i.e., $n_x = \mathbf{n}(\mathfrak{B})$, if the polynomials a and b in (8) are left-coprime. This means that we also have $\mathbf{L}(\mathfrak{B}) = \mathbf{n}(\mathfrak{B})$. For MIMO systems, i.e., $n_u, n_y > 1$, this is generally not the case as $\mathbf{L}(\mathfrak{B}) \leq \mathbf{n}(\mathfrak{B})$. In the MIMO case, a minimal realization of (9) from (11) can always be obtained by means of moment matching⁴ [31] or an LPV Kalman decomposition [32]. With both methods, the minimal realization of (9) is obtained using a *constant* projection matrix that projects the state to a lower dimension. This means that the resulting (reduced) $A(p), \dots, D(p)$ will still have static scheduling dependence and the *same* functional dependence (e.g., C is still scheduling independent).

B. Complexity and dimension of behaviors

From Section III-A, we recovered the integers $\mathbf{m}(\mathfrak{B})$, $\mathbf{L}(\mathfrak{B})$, and $\mathbf{n}(\mathfrak{B})$ that are a measure for the complexity of the behavior \mathfrak{B} of an LPV-SA system $\Sigma \in \Sigma_{n_p, n_w}$. In line with [26], we characterize the complexity by the triple

$$\mathbf{c}(\mathfrak{B}) = (\mathbf{m}(\mathfrak{B}), \mathbf{L}(\mathfrak{B}), \mathbf{n}(\mathfrak{B})), \quad (13)$$

where $\mathbf{m}(\mathfrak{B})$ is the number of inputs, $\mathbf{L}(\mathfrak{B})$ is the minimal lag of the system, and $\mathbf{n}(\mathfrak{B})$ is the order of \mathfrak{B} .

With these integer invariants defined, we will now formulate one of the key ingredients required for the solution to Problem 2. More specifically, we now show that for an $L \geq \mathbf{L}(\mathfrak{B})$, the dimension of $\mathfrak{B}_p|_{[1,L]}$ with $p_{[1,L]} \in \mathfrak{B}_{\mathbb{P}}|_{[1,L]}$ is equal to $\mathbf{n}(\mathfrak{B}) + \mathbf{m}(\mathfrak{B})L$.

Lemma 1 (Dimension of $\mathfrak{B}_p|_{[1,L]}$). *Consider an LPV-SA system $\Sigma \in \Sigma_{n_p, n_w}$ with behavior \mathfrak{B} and complexity $\mathbf{c}(\mathfrak{B})$. Given any $p_{[1,L]} \in \mathfrak{B}_{\mathbb{P}}|_{[1,L]}$. Then, $\dim(\mathfrak{B}_p|_{[1,L]}) = \mathbf{n}(\mathfrak{B}) + \mathbf{m}(\mathfrak{B})L$ if and only if $L \geq \mathbf{L}(\mathfrak{B})$.*

Proof. See Appendix I. ■

This result will allow us to prove what we call the LPV Fundamental Lemma for the class of LPV-SA systems Σ_{n_p, n_w} .

⁴See also the implementation in LPVCORE [30].

Note that we have presented a version of Lemma 1 in the meromorphic context in preliminary work [21, Cor. 1]. In this paper, we prove this result in the context of the class of LPV-SA systems.

We now have all the ingredients for the formulation of a data-driven representation of the finite-horizon behavior of LPV-SA systems, and thus to solve Problems 1 and 2.

IV. DATA-DRIVEN REPRESENTATION OF LPV-SA SYSTEMS

This section presents the first part of our main result, which is the data-driven characterization of LPV-SA systems. First, a representation of the finite-horizon behavior of LPV-SA systems based on a given data set \mathcal{D}_{N_d} is introduced, providing a solution to Problem 1. Next, we derive a GPE condition to test whether \mathcal{D}_{N_d} is ‘rich’ enough to fully characterize the finite-horizon behavior, providing a solution to Problem 2. We conclude this section with a note on input design.

A. Data-driven representation

We formulate a data-driven representation of $\mathfrak{B}_p|_{[1,L]}$ that is valid for any $p \in \mathfrak{B}_{\mathbb{P}}|_{[1,L]}$ by means of embedding the behavior represented by the kernel representation into an LTI realization, whose behavior is constrained by a scheduling-dependent kernel constraint. Isolating a single term in the polynomial corresponding to the kernel representation (1)

$$\left(r_{i,0} + \sum_{j=1}^{n_p} r_{i,j} p_j(k+i) \right) w(k+i), \quad (14)$$

reveals that we can write the individual terms of (1), i.e., (14), in terms of the *auxiliary signal*

$$w'(k+i) = \begin{bmatrix} 1 \\ p(k+i) \end{bmatrix} \otimes w(k+i) \in \mathbb{R}^{(1+n_p)n_w}, \quad (15)$$

such that

$$(r_{i,0} + \sum_{j=1}^{n_p} r_{i,j} p_j(k+i)) w(k+i) = \underbrace{\begin{bmatrix} r_{i,0} & \cdots & r_{i,n_p} \end{bmatrix}}_{r'_i} w'(k+i). \quad (16)$$

The definition of this auxiliary signal w' allows us to embed the behavior associated with a kernel representation with shifted-affine dependency as an *LTI representation*. Intuitively, this LTI embedding treats the $n_p n_w$ variables of $p(k) \otimes w(k)$ as additional inputs, i.e., free variables, even if they are not. The LTI embedding of (1) gives

$$R'(q)w' = \sum_{i=0}^{n_r} r'_i q^i w' = 0, \quad (17)$$

with the behavior

$$\mathfrak{B}' = \{w' : \mathbb{T} \rightarrow \mathbb{R}^{(1+n_p)n_w} \mid (17) \text{ holds}\}. \quad (18)$$

The main difference between \mathfrak{B} and \mathfrak{B}' is that the entries of w' are *not* independent from each other in the original LPV representation (1), while in the LTI embedding (17) this interdependency is *ignored*. Hence, for some $w' \in \mathfrak{B}'$ there might not exist a pair $(w, p) \in \mathfrak{B}$ such that $w'(k) =$

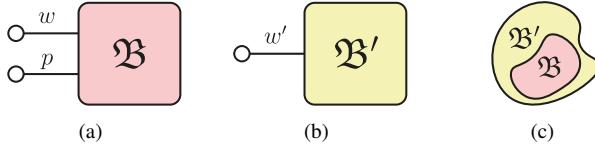


Fig. 2: Illustration of the (a) LPV system versus (b) its LTI embedding and (c) their corresponding behaviors. The behavior of the LTI embedding over-approximates the behavior of the LPV system.

$\text{col}(w(k), p(k) \otimes w(k))$ for all k , implying that $\mathfrak{B} \subset \mathfrak{B}'$. The resulting behavior of the LTI embedding thus *over-approximates* the behavior of the LPV system, see Fig. 2c for illustration. We will counteract this over-approximation in the next paragraph, to arrive at an exact data-driven LPV representation of $\mathfrak{B}_p|_{[1,L]}$. As the introduced auxiliary signals $p(k) \otimes w(k) \in \mathbb{R}^{n_p n_w}$ are considered to be free, we have that

$$\mathbf{m}(\mathfrak{B}') = \mathbf{m}(\mathfrak{B}) + n_p n_w. \quad (19)$$

Additionally, this consideration implies that (17) is an LTI system on which we can apply the existing results on data-driven LTI representations, i.e., Willems' Fundamental Lemma and its associated results [1], [3]. For a given data set $\mathcal{D}_{N_d} = \check{w}'_{[1,N_d]}$, these results provide that

$$\text{image}(\mathcal{H}_L(\check{w}'_{[1,N_d]})) \subseteq \mathfrak{B}'|_{[1,L]}, \quad (20)$$

where equality holds if and only if the GPE (*generalized persistence of excitation*) condition for LTI systems [3, Cor. 19] holds, i.e.,

$$\text{rank}(\mathcal{H}_L(\check{w}'_{[1,N_d]})) = \mathbf{n}(\mathfrak{B}) + \mathbf{m}(\mathfrak{B}')L. \quad (21)$$

Given that (21) holds, we thus have that for any $w'_{[1,L]} \in \mathfrak{B}'|_{[1,L]}$, there exists a $g \in \mathbb{R}^{N_d-L+1}$ such that

$$\mathcal{H}_L(\check{w}'_{[1,N_d]}g) = \text{vec}(w'_{[1,L]}) \quad (22)$$

holds. Hence, the equality (22) serves as a data-driven representation of the extended behavior \mathfrak{B}' of the LTI embedding. We will now take back into account the previously ignored interdependences in w' , and thus counteract the aforementioned over-approximation.

Let us introduce the signal $w^p_{[1,N_d]} = (p(k) \otimes w(k))_{k=1}^{N_d}$ such that $w'_{[1,N_d]} = \text{col}(w_{[1,N_d]}, w^p_{[1,N_d]})$. This allows us to write (22), after a permutation of the rows, as

$$\begin{bmatrix} \mathcal{H}_L(\check{w}'_{[1,N_d]}) \\ \mathcal{H}_L(\check{w}^p_{[1,N_d]}) \end{bmatrix} g = \begin{bmatrix} \text{vec}(w_{[1,L]}) \\ \text{vec}(w^p_{[1,L]}) \end{bmatrix}. \quad (23)$$

As observed in [22], $\text{vec}(w^p_{[1,L]}) = \mathcal{P}^{n_w} \text{vec}(w_{[1,L]})$, where $\mathcal{P}^{n_w} = p_{[1,L]} \odot I_{n_w}$. Hence,

$$\text{vec}(w^p_{[1,L]}) = \mathcal{P}^{n_w} \text{vec}(w_{[1,L]}) = \mathcal{P}^{n_w} \mathcal{H}_L(\check{w}'_{[1,N_d]})g. \quad (24)$$

Any g satisfying (24), respects the underlying dynamic structure w.r.t. the scheduling in $w^p_{[1,L]}$, i.e., it serves as a *restriction* on the behavior of the LTI embedding that *eliminates* the aforementioned over-approximation. This restriction becomes visible when we substitute (24) into (23), and

move ' $\mathcal{P}^{n_w} \mathcal{H}_L(\check{w}'_{[1,N_d]})g$ ' to the left-hand side of (23):

$$\begin{bmatrix} \mathcal{H}_L(\check{w}'_{[1,N_d]}) \\ \mathcal{H}_L(\check{w}^p_{[1,N_d]}) - \mathcal{P}^{n_w} \mathcal{H}_L(\check{w}'_{[1,N_d]}) \end{bmatrix} g = \begin{bmatrix} \text{vec}(w_{[1,L]}) \\ 0 \end{bmatrix}. \quad (25)$$

We can notice that the first block-row in (25) characterizes the LTI part of the LPV system (associated with $r_{i,0}$), while the second block-row provides a restriction on the g vectors that provide trajectories $w_{[1,L]}$ from the linear combination of the columns of $\mathcal{H}_L(\check{w}'_{[1,N_d]})$. This restriction is not only dependent on the information encoded in $(\check{w}'_{[1,N_d]}, \check{p}_{[1,N_d]})$, i.e., \mathcal{D}_{N_d} , but also on the scheduling signal $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$ associated with $w_{[1,L]}$. Hence, the left-hand side of (25) provides us with a *data-driven* characterization of $\mathfrak{B}_p|_{[1,L]}$, i.e., a characterization of all $w_{[1,L]}$ trajectories that correspond to the scheduling trajectory $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$.

Given sufficiently rich data, we now established that all possible $w_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$ are characterized by vectors g that satisfy (25), i.e., vectors g that are both in the row space of $\mathcal{H}_L(\check{w}'_{[1,N_d]})$ and the kernel of $\mathcal{H}_L(\check{w}^p_{[1,N_d]}) - \mathcal{P}^{n_w} \mathcal{H}_L(\check{w}'_{[1,N_d]})$ for a given scheduling trajectory $p_{[1,L]}$. Hence, by defining

$$\mathcal{N}_p = \text{kernel} \left(\mathcal{H}_L(\check{w}^p_{[1,N_d]}) - \mathcal{P}^{n_w} \mathcal{H}_L(\check{w}'_{[1,N_d]}) \right), \quad (26)$$

we have that

$$\text{image}(\mathcal{H}_L(\check{w}'_{[1,N_d]}) \mathcal{N}_p) \subseteq \mathfrak{B}_p|_{[1,L]}. \quad (27)$$

We now established an *exact* data-driven representation of the set of $w_{[1,L]}$ sequences associated with $p_{[1,L]}$, corresponding to Problem 1 (Contribution C1). Next, we will establish a condition on \mathcal{D}_{N_d} that guarantees (27) to hold with equality, providing the solution to Problem 2 (Contribution C2).

B. The LPV fundamental lemma

In this section, we provide a necessary and sufficient condition verifiable from the data \mathcal{D}_{N_d} that ensures whether (27) holds with equality, providing Contribution C2. We establish this by showing that the dimension of $\text{image}(\mathcal{H}_L(\check{w}'_{[1,N_d]}) \mathcal{N}_p)$ is strongly linked to the dimensionality of $\mathfrak{B}_p|_{[1,L]}$. This results in an so-called ‘‘identifiability condition’’ for LPV-SA systems, i.e., a GPE condition that is analogous to the condition put forward in [3]. In other words, we prove the necessary and sufficient conditions that the data in \mathcal{D}_{N_d} should satisfy to be able to characterize the full $\mathfrak{B}_p|_{[1,L]}$ for a given $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$.

Theorem 1 (LPV-SA Fundamental Lemma). *Given a data set $\mathcal{D}_{N_d} \in \mathfrak{B}|_{[1,N_d]}$ from an LPV-SA system $\dot{\Sigma} \in \Sigma_{n_p, n_w}$. Construct \mathcal{N}_p for some $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$ as in (26). For $L \geq L(\mathfrak{B})$, the following statements are equivalent:*

i). *For all $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$,*

$$\mathfrak{B}_p|_{[1,L]} = \text{image}(\mathcal{H}_L(\check{w}'_{[1,N_d]}) \mathcal{N}_p), \quad (28)$$

ii). *For all $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$,*

$$\text{rank}(\mathcal{H}_L(\check{w}'_{[1,N_d]}) \mathcal{N}_p) = \mathbf{n}(\mathfrak{B}) + \mathbf{m}(\mathfrak{B})L, \quad (29)$$

iii). *For any $(w_{[1,L]}, p_{[1,L]}) \in \mathfrak{B}|_{[1,L]}$, there exists a vector $g \in \mathbb{R}^{N_d-L+1}$ such that (25) holds,*

iv). The following rank condition is satisfied:

$$\text{rank} \left(\begin{bmatrix} \mathcal{H}_L(\check{w}_{[1,N_d]}) \\ \mathcal{H}_L(\check{w}_{[1,N_d]}^p) \end{bmatrix} \right) = \mathbf{n}(\mathfrak{B}) + (\mathbf{m}(\mathfrak{B}) + n_p n_w)L. \quad (30)$$

Proof. See Appendix II. \blacksquare

Item (ii) in Theorem 1 provides a necessary and sufficient dimensionality condition on the data-driven representation of $\mathfrak{B}_p|_{[1,L]}$ that is verifiable from the data set \mathcal{D}_{N_d} . Specifically, the data can fully represent $\mathfrak{B}_p|_{[1,L]}$ for any $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$ if and only if (29) holds for all $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$. Although, (29) seems to result in an infinite test over all possible $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$, through the LTI embedding, we show with Item (iv) that this reduces to a single, simple rank test on the left-hand side of (23), which is only composed from the given data in \mathcal{D}_{N_d} . We will refer to (30) as the LPV-GPE condition. The LPV-GPE condition also provides a lower bound for N_d :

$$N_d \geq (1 + n_w n_p + \mathbf{m}(\mathfrak{B}))L + \mathbf{n}(\mathfrak{B}) - 1, \quad (31)$$

i.e., the minimum number of samples in \mathcal{D}_{N_d} required to represent $\mathfrak{B}_p|_{[1,L]}$ for an arbitrary $p_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$. It is important to highlight that the LPV-GPE condition provides no separate PE condition on the used scheduling trajectory $\check{p}_{[1,N_d]}$ or input signal $\check{u}_{[1,N_d]}$ in the data set, only on the joint collection of input, scheduling, and output signals in the data-dictionary (through (30)). In the next section, we provide a discussion on generating a \mathcal{D}_{N_d} that satisfies (30).

C. Input design

What makes Theorem 1 different from the original (LPV) Fundamental Lemma in [1], [21] is that we now have a rank condition on the Hankel matrices involving trajectories of w and p , while in [1], the rank condition is only on the input signal u , specifically, $\text{rank}(\mathcal{H}_{L+\mathbf{n}(\mathfrak{B})}(\check{u}_{[1,N_d]})) = \mathbf{m}(\mathfrak{B})(L + \mathbf{n}(\mathfrak{B}))$. The latter yields an *input design* condition, which allows you to *a priori* design an experiment for the construction of a data-driven representation, without taking the output of the system itself into account. It would be tempting to adopt this condition for the LPV-SA case, i.e., by taking this rank condition on the Hankel matrix of $[\check{u}^p]$. This would lead to the condition that if

$$\text{rank} \left(\begin{bmatrix} \mathcal{H}_{L+\mathbf{n}(\mathfrak{B})}(\check{u}_{[1,N_d]}) \\ \mathcal{H}_{L+\mathbf{n}(\mathfrak{B})}(\check{u}_{[1,N_d]}^p) \end{bmatrix} \right) = \mathbf{m}(\mathfrak{B})(1 + n_p)(\mathbf{n}(\mathfrak{B}) + L), \quad (32)$$

then (28) holds. In the following counter example, we show that this is, unfortunately, not the case.

Example 2. Consider an LPV-SA system $\Sigma \in \Sigma_{n_p, n_w}$ with the LPV-IO representation:

$$y(k) + (1 + p(k-1))y(k-1) = u(k) + p(k-1)u(k-1),$$

which has shifted-affine scheduling dependence. Note that $\mathbf{n}(\mathfrak{B}) = \mathbf{m}(\mathfrak{B}) = \mathbf{L}(\mathfrak{B}) = n_p = 1$ and $n_w = 2$ for this particular system. We compute a data-driven representation

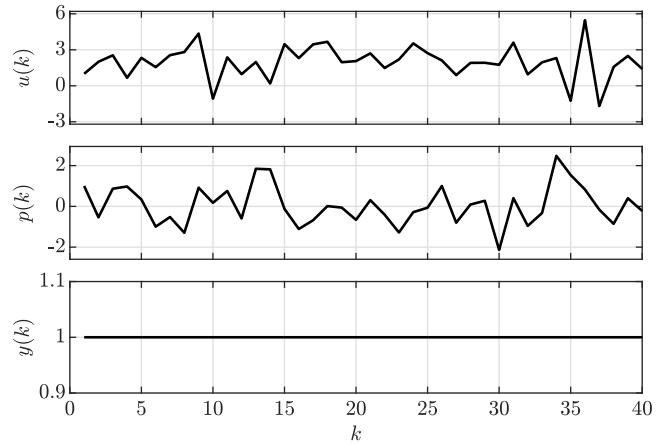


Fig. 3: Input, scheduling, and output sequences of \mathcal{D}_{N_d} in Example 2.

of Σ for $L = 10$. For this, we generate a \mathcal{D}_{N_d} with $N_d = 40$, according to (31). With $y(0) = u(0) = p(0) = 1$, we apply an i.i.d. scheduling signal $p_{[1,N_d]}$ with $p(k) \sim \mathcal{N}(0, 1)$ and an input that is constructed as:

$$u(k) = p(k-1)(1 - u(k-1)) + 2,$$

to the system. Note that this input can in fact be seen as a control policy to regulate the system to $y(k) = 1$ for any scheduling sequence. The resulting input and scheduling sequences are shown in Fig. 3. By looking at these signals, one could already argue that they are persistently exciting, and, indeed, computing the rank in condition (32) gives a rank of 22 (note that $\mathbf{m}(\mathfrak{B})(1 + n_p)(L + \mathbf{n}(\mathfrak{B})) = 22$). However, when inspecting the output response, we see that $y(k) = 1$ for all $k = 1, \dots, N_d$, which is due to the construction of $u(k)$. This means that (29) and thus (28) will never be satisfied, showing that directly adopting the *input-design* condition from [1] does not apply for systems of the class Σ_{n_p, n_w} . Applying the condition on the LTI embedding (17) with inputs $\text{col}(u, w^p)$, gives the correct conclusion, i.e.,

$$\text{rank} \left(\begin{bmatrix} \mathcal{H}_{L+\mathbf{n}(\mathfrak{B})}(\check{u}_{[1,N_d]}) \\ \mathcal{H}_{L+\mathbf{n}(\mathfrak{B})}(\check{u}_{[1,N_d]}^p) \\ \mathcal{H}_{L+\mathbf{n}(\mathfrak{B})}(\check{y}_{[1,N_d]}) \end{bmatrix} \right) = \underbrace{(\mathbf{m}(\mathfrak{B}) + n_w n_p)(L + \mathbf{n}(\mathfrak{B}))}_{\mathbf{m}(\mathfrak{B}')}, \quad (33)$$

should hold in order for (28) to hold. Computing the rank as in (33) with the obtained data set gives 24, while $(\mathbf{m}(\mathfrak{B}) + n_w n_p)(L + \mathbf{n}(\mathfrak{B})) = 33$, concluding that the data cannot represent the considered system for $L = 10$. \blacktriangleleft

From Example 2, we see that, in order to have a condition on the input (and scheduling) that *a priori* guarantees (28) to hold, we need a condition on the design of $(\check{u}_{[1,N_d]}, \check{p}_{[1,N_d]})$ which guarantees that the resulting $(\text{col}(\check{u}_{[1,N_d]}, \check{y}_{[1,N_d]}), \check{p}_{[1,N_d]}) \in \mathfrak{B}|_{[1,N_d]}$ will satisfy (33). We currently do not have a systematic solution to this problem. Hence, we currently only have methods that can *a posteriori* verify whether the data satisfies the LPV-GPE condition. The development of input (and scheduling) design conditions that *a priori* guarantee

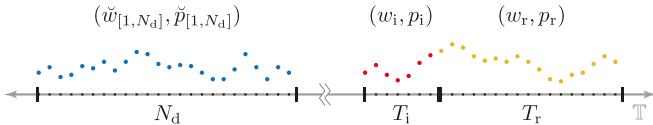


Fig. 4: Schematic representation of the simulation problem: With a length- T_i initial trajectory (w_i, p_i) (depicted by the red data points), determine the response (w_r, p_r) (depicted by the yellow data points) using only a given data set $\mathcal{D}_{N_d} = (\check{w}_{[1,N_d]}, \check{p}_{[1,N_d]})$ (depicted by the blue data points).

satisfaction of (30) is an important and interesting topic for future research.

V. DATA-DRIVEN SIMULATION

In this section, we consider the solution to the simulation problem in a data-driven setting in terms of Problem 3. We want to emphasize that, although the data-driven simulation has been used already in the LPV setting, see [21]–[23], it has never been worked out in detail from a theoretical perspective. Therefore, we formally work out the solution to the simulation problem in this section as a generalization of the LTI data-driven simulation that is presented in [33].

A. The LPV data-driven simulation problem

The general form of the simulation problem is: given an LPV-SA system $\Sigma \in \Sigma_{n_p, n_w}$ with IO partition $w = \text{col}(u, y)$, starting from an initial condition, find the response y_r of Σ to (u_r, p_r) such that $(\text{col}(u_r, y_r), p_r) \in \mathfrak{B}|_{[1, T_r]}$.

Classically, the initial condition is characterized by an initial state x of an LPV-SS representation of \mathfrak{B} , but equivalently such an initial condition can be (uniquely) expressed in terms of an *initial trajectory*. Hence, this provides a way to formalize the *data-driven simulation problem*, i.e., Problem 3, as follows:

Problem 4. Given a data set $\mathcal{D}_{N_d} \in \mathfrak{B}|_{[1, N_d]}$ from an LPV-SA system $\check{\Sigma} \in \Sigma_{n_p, n_w}$, an input-scheduling pair $(u_r, p_r) \in (\mathbb{R}^{n_u} \times \mathbb{P})^{[1, T_r]}$ and an initial trajectory $(w_i, p_i) \in \mathfrak{B}|_{[1, T_i]}$. Find a response y_r of $\check{\Sigma}$ to the input-scheduling pair $(u_r, p_r) \in (\mathbb{R}^{n_u} \times \mathbb{P})^{[1, T_r]}$, such that $(w_i, p_i) \wedge (\text{col}(u_r, y_r), p_r) \in \mathfrak{B}|_{[1, T_i + T_r]}$.

The simulation problem is also illustrated in Fig. 4.

B. Trajectory-based initial condition

First, we formally show that the initial condition x of the response (w_r, p_r) can be uniquely determined with a length- T_i initial trajectory if T_i is larger than $L(\mathfrak{B})$.

Lemma 2 (Initial state characterization). *Consider an LPV-SA system $\Sigma \in \Sigma_{n_p, n_w}$ with behavior \mathfrak{B} . Given $(w_i, p_i) \in (\mathbb{R}^{n_w} \times \mathbb{P})^{[1, T_i]}$. If $T_i \geq L(\mathfrak{B})$ and*

$$(w_i, p_i) \wedge (w_r, p_r) \in \mathfrak{B}|_{[1, T_i + T_r]}, \quad (34)$$

then, the initial condition of (w_r, p_r) is uniquely expressed in terms of (w_i, p_i) . Equivalently, for a minimal LPV-SS representation of Σ , there exists a unique $x \in \mathbb{R}^{n(\mathfrak{B})}$ that serves as the initial condition of (w_r, p_r) .

Algorithm 1 LPV data-driven simulation

Requires A data set $\mathcal{D}_{N_d} \in \mathfrak{B}|_{[1, N_d]}$, an initial trajectory $(w_i, p_i) \in \mathfrak{B}|_{[1, T_i]}$, and an input-scheduling trajectory $(u_r, p_r) \in \pi_{u,p} \mathfrak{B}|_{[1, T_r]}$.

1: Compute a g that satisfies

$$\begin{bmatrix} \mathcal{H}_{T_i}(\check{w}_{[1, N_d - T_r]}) \\ \mathcal{H}_{T_r}(\check{u}_{[T_i + 1, N_d]}) \\ \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}^p) - \mathcal{P}_{i,r}^{n_w} \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}) \end{bmatrix} g = \begin{bmatrix} \text{vec}(w_i) \\ \text{vec}(u_r) \\ 0 \end{bmatrix}, \quad (35)$$

with $\bar{T} = T_i + T_r$ and $\mathcal{P}_{i,r}^{n_w} = (p_i \wedge p_r) \odot I_{n_w}$.

2: Compute y_r via

$$\text{vec}(y_r) = \mathcal{H}_{T_r}(\check{y}_{[T_i + 1, N_d]})g. \quad (36)$$

Outputs y_r

Proof. See Appendix III. ■

The result of Lemma 2 provides us with a condition ($T_i \geq L(\mathfrak{B})$) that ensures a unique⁵ response to an input-scheduling pair. In the next section, we give a solution to the LPV data-driven simulation problem.

C. The LPV data-driven simulation algorithm

To satisfy $(w_i, p_i) \wedge (\text{col}(u_r, y_r), p_r) \in \mathfrak{B}|_{[1, T_i + T_r]}$, we know from Theorem 1 that (29) and thus (30) should hold for $L = T_i + T_r$. This allows to write

$$\begin{bmatrix} \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}) \\ \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}^p) - \mathcal{P}_{i,r}^{n_w} \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}) \end{bmatrix} g = \begin{bmatrix} \text{vec}(w_i) \\ \text{vec}(u_r) \\ \text{vec}(y_r) \\ 0 \end{bmatrix},$$

where $\bar{T} = T_i + T_r$ and $\mathcal{P}_{i,r}^{n_w} = (p_i \wedge p_r) \odot I_{n_w}$. For a given w_i , u_r , and scheduling sequence $p_i \wedge p_r$, this is a linear set of equations in the unknowns g and y_r . Partitioning of the Hankel matrices on the left-hand side provides Algorithm 1, accomplishing LPV data-driven simulation. The LPV generalizations of the special cases of LTI data-driven simulation discussed in [33], e.g., zero input response, zero initial condition response, impulse response, etc., directly follow from the presented derivation of the LPV data-driven simulation algorithm. This is also the case for, e.g., recursive implementation of Algorithm 1 to simulate for $T_r \rightarrow \infty$, data-driven simulation for LPV embeddings of nonlinear systems via iterative scheduling refinement,⁶ or data-based scheduling estimation. While these topics are interesting, due to the sake of space, they are not discussed in detail in this paper.

⁵The formal problem formulation in Section V-A does *not* require finding a unique response for y_r , which is, however, often desired.

⁶In the case of nonlinear systems, the scheduling is generally dependent on w through a so-called scheduling map. In [34], an iterative procedure in the context of model predictive control is used to obtain the w -dependent scheduling signal, similar to the approach used in sequential quadratic programming. See also Example 4 for application of the iterative scheme in a nonlinear data-driven control setting.

TABLE I: Parameters of the MSD system.

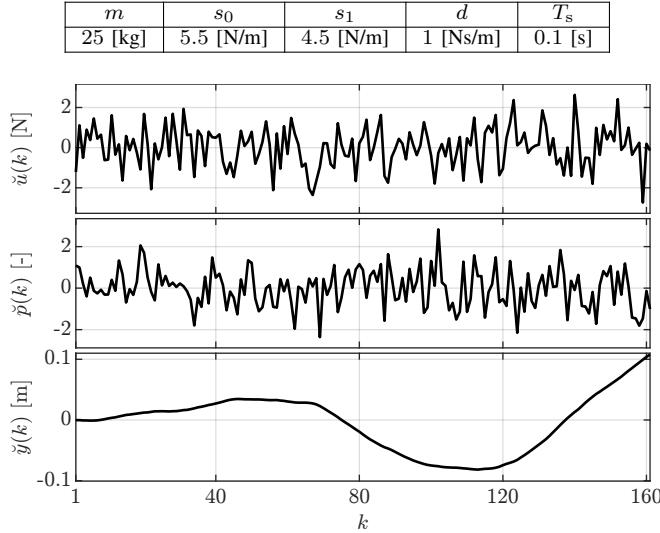


Fig. 5: Data-dictionary measured from the LPV MSD system with $N_d = 161$.

VI. EXAMPLES

To illustrate the validity and effectiveness of the data-driven representations, we demonstrate their capabilities on an LPV system and an LPV embedding of a nonlinear system in a simulation and control scenario.

Example 3. For this first example, we use the MSD system presented in Example 1 with parameters given in Table I. Note that the system is SISO and $L(\mathcal{B}) = n(\mathcal{B}) = 2$. In this example, we want to perform a data-driven simulation of this system for $T_r = 35$ samples (corresponding to 3.5 seconds), without having access to (5) or its parameters; only a measured data set from the system is available. We consider three cases:

- Case 1: Conditions of Theorem 1 hold and $T_i \geq L(\mathcal{B})$;
- Case 2: Conditions of Theorem 1 do not hold and $T_i \geq L(\mathcal{B})$;
- Case 3: Conditions of Theorem 1 hold and $T_i < L(\mathcal{B})$.

For the Cases 1 and 2, $T_i = 5$ is chosen, while, for Case 3, $T_i = 1$. We generate a data-dictionary \mathcal{D}_{N_d} of length $N_d = (1 + n_w n_p + m(\mathcal{B}))L + n(\mathcal{B}) - 1 = 161$, cf. (31), and take, for Case 2, $N_d = 151$ by disregarding the last 10 samples in \mathcal{D}_{N_d} . The data-dictionary used in this example is shown in Fig. 5, where $\check{u}(k) \sim \mathcal{N}(0, 1)$ and $\check{p}(k) \sim \mathcal{N}(0, 1)$. With (30), it is verified that \mathcal{D}_{N_d} satisfies the LPV-GPE condition and can represent the behavior on the horizon $T_i + T_r$ for Cases 1 and 3, while the LPV-GPE condition does not hold for Case 2.

We now solve Algorithm 1 for the aforementioned three cases. By solving (35) in the least-squares sense and observe that, as expected, the following norm

$$\left\| \begin{bmatrix} \mathcal{H}_{T_i}(\check{w}_{[1, N_d - T_r]}) \\ \mathcal{H}_{T_r}(\check{u}_{[T_i + 1, N_d]}) \\ \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}^{\check{p}}) - \mathcal{P}_{i,r}^{n_w} \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}) \end{bmatrix} g - \begin{bmatrix} \text{vec}(w_i) \\ \text{vec}(u_r) \\ 0 \end{bmatrix} \right\|_2$$

is zero for Cases 1 and 3, while, for Case 2, it is 0.58. Computing the simulated outputs with (36) for all the considered cases gives the results in Fig. 6, which are compared to a model-based simulation of (5) shown in red. In Case 1, all the

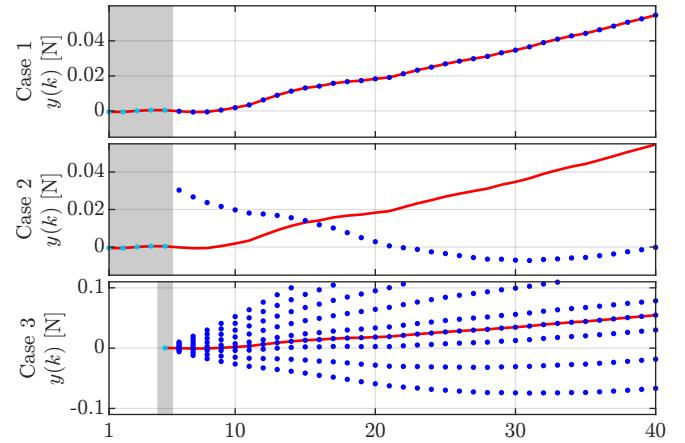


Fig. 6: Simulation results for Cases 1–3. The red solid line is the model-based simulation, the initial trajectory is indicated with the light-blue dotted line and the gray shaded area, while the blue dotted line represents the data-driven simulation result. For Case 1 the model-based and data-based simulations coincide. Case 2 results in a simulated output that is not connected to the initial trajectory and hence does not coincide with the model-based simulation. The data-driven simulation results for Case 3 show a continuation of the initial trajectory, but the simulated output trajectory is not unique which is illustrated by plotting several valid solutions for y_r .

conditions for LPV data-driven simulation are satisfied, and indeed the model-based and data-based simulations coincide. The simulation result for Case 2 shows that y_r is both not connected to the initial trajectory and not coinciding with the model-based simulation, i.e., the behavior, which contains the true trajectory, is not represented by the data in the smaller \mathcal{D}_{N_d} . The solution to (35) with Case 3 results in a larger space for the valid g vectors and thus a set of possible y_r trajectories. This is because with $T_i < L(\mathcal{B})$, the problem (35) is under-determined, i.e., there are infinitely many solutions for y_r for which $(\text{col}(u_i, y_i), p_i) \wedge (\text{col}(u_r, y_r), p_r) \in \mathcal{B}|_{[1, T_i + T_r]}$. We have illustrated this by plotting a number of valid solutions for y_r , all of which are a valid continuation of the initial trajectory. ▀

Example 4. In this next example, we apply our methods on a nonlinear system that is embedded in an LPV form. Consider the nonlinear system

$$\begin{aligned} y(k) + (0.2 - 0.4 \tanh(y(k-1)))y(k-1) \\ + \tanh(y(k-2))y(k-2) = 1.2u(k-1) \\ + 0.4 \sin(u(k-1))e^{-y^2(k-1)} \\ + (1 + 0.6 \tanh(y(k-2)))y(k-2), \quad (37) \end{aligned}$$

and define $p := \psi(u, y) = [\tanh(y) \ \text{sinc}(u)e^{-y^2}]^\top$. This *scheduling map* defines an LPV embedding of (37) that has the form (1). We will now use Algorithm 1 in a control setting (by seeing w_r as a decision variable) to achieve direct data-driven feedforward control of the nonlinear system (37) under the assumption that we know ψ (see [20] on how to overcome this assumption). To achieve this, we will make use of Footnote 6. We want to regulate the system (37) to its origin in $T_r = 30$ time-steps from an arbitrary initial trajectory of length $T_i = 3$, without knowing anything about the system, except a given

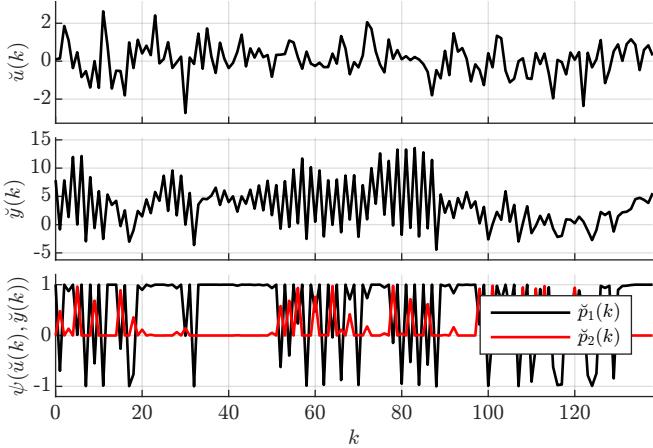


Fig. 7: Input, output, and scheduling sequences of \mathcal{D}_{N_d} in Example 4. The scheduling p is defined by the scheduling map ψ .

Algorithm 2 Data-driven control for nonlinear systems

Requires A data set $\mathcal{D}_{N_d} \in \mathfrak{B}|_{[1, N_d]}$ for which (30) holds, an initial trajectory $(w_i, p_i) \in \mathfrak{B}|_{[1, T_i]}$, and an initial guess for p_r .

1: **Repeat**

- 2: Let $\mathcal{P}_{i,r}^{n_w} \leftarrow (p_i \wedge p_r) \odot I_{n_w}$,
- 3: Solve (38) and obtain (u_r, y_r)
- 4: Update $p_r \leftarrow \psi(u_r, y_r)$

5: **Until** p_r has converged

Outputs u_r

data set recorded a priori. To obtain the data set \mathcal{D}_{N_d} , we generate $N_d = 139$ data samples, cf. (31), by applying an input $u(k) \sim \mathcal{N}(0, 1)$ to (37). The resulting trajectories are shown in Fig. 7. A posteriori verification of (30) gives that \mathcal{D}_{N_d} can represent $\mathfrak{B}|_{[1, T_i + T_r]}$.

Now, by exploiting Algorithm 1, we formulate the solution of a predictive control problem, i.e., computing the required input sequence on a given horizon to drive the nonlinear system from arbitrary initial conditions to the origin. Specifically, we iteratively solve the following quadratic program:

$$\min_g (*)^\top Q \text{vec}(y_r) + (*)^\top R \text{vec}(u_r) \quad (38a)$$

$$\text{s.t. } \begin{bmatrix} \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}) \\ \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}^{\check{p}}) - \mathcal{P}_{i,r}^{n_w} \mathcal{H}_{\bar{T}}(\check{w}_{[1, N_d]}) \end{bmatrix} g = \begin{bmatrix} \text{vec}(w_i) \\ \text{vec}(u_r) \\ \text{vec}(y_r) \\ 0 \end{bmatrix}, \quad (38b)$$

where in each iteration, after solving (38), we update $\mathcal{P}_{i,r}^{n_w}$ using the scheduling map ψ applied on the solution (u_r, y_r) .

This procedure is summarized in Algorithm 2. In [35] it is shown that with some additional minor modifications, this sequence of quadratic programs has a local contraction property under assumptions similar to those for standard sequential quadratic programming methods.

In this example, we choose $Q = R = I$ and take a zero-trajectory as initial guess for p_r . We solve Algorithm 2 for the

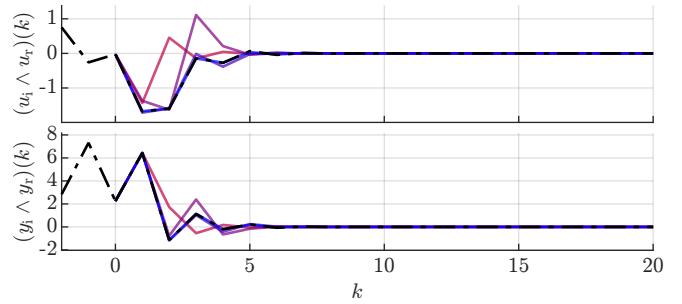


Fig. 8: Input and output sequences for different iterations in Algorithm 2 depicted with blue (less transparent lines correspond to a higher iteration number). The final input trajectory obtained from Algorithm 2 and the resulting output trajectory from the true system are depicted in black.

initial trajectory

$$w_i = ([0.75], [-0.26], [-0.03]), \\ p_i = ([0.99], [0.99], [0.978]),$$

and took $\|p_r^{n-1} - p_r^n\| < 10^{-6}$ as the stopping criterion for Step 5 of Algorithm 2 with p_r^n the updated scheduling sequence after the n^{th} iteration. The trajectory for p_r converged in 12 steps. The difference between y_r coming from (38) and the true simulation with the implemented optimal u_r (in the 2-norm) is $1.5 \cdot 10^{-7}$. We have plotted the results in Fig. 8. This figure also shows (u_r^n, y_r^n) , i.e., the optimized trajectories for iteration n in Step 3 of Algorithm 2. The iterations are plotted in the color spectrum from red (small n) and blue (large n). These results show that the simulated trajectories of converge to the true solution of the nonlinear system relatively fast, by simply solving a number of simple quadratic programs. This highlights how the derived results can be applied to establish efficient direct data-driven analysis and control methods for nonlinear systems. ▀

VII. CONCLUSIONS

In this paper, we derived foundational results for data-driven LPV analysis and control of LPV systems in terms of establishing a computable data-driven representation of LPV systems together with a generalized form of a persistency of excitation condition. The latter condition characterizes when the data is sufficiently informative to describe the underlying behavior of the system via such a data-driven representation. Additionally, we provided a formal solution to the LPV data-driven simulation problem to compute responses of LPV systems for a given input and scheduling trajectory when no model based knowledge is available for the system except previously recorded data. We demonstrated validity of our results through various examples, also showing applicability of the proposed approaches for data-driven handling of nonlinear systems.

In our work, we restricted the scope to LPV-SA systems, i.e., systems that can be represented by an LPV-IO representation with shifted-affine scheduling dependence, which is a practically useful class of systems. However, it is an important objective of future research to generalize the results to other

useful scheduling dependency classes and to handling noisy data sets. Moreover, we consider the formulation of conditions on the input and scheduling trajectories that allow for experiment design with a priori guarantees on the satisfaction of (30) an important topic for future research.

REFERENCES

- [1] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. M. De Moor, “A note on persistency of excitation,” *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.
- [2] I. Markovsky and F. Dörfler, “Behavioral systems theory in data-driven analysis, signal processing, and control,” *Annual Reviews in Control*, vol. 52, pp. 42–64, 2021.
- [3] ——, “Identifiability in the behavioral setting,” *IEEE Transactions on Automatic Control*, vol. 68, no. 3, pp. 1667–1677, 2022.
- [4] P. Rapisarda, H. J. van Waarde, and M. K. Camlibel, “Orthogonal polynomial bases for data-driven analysis and control of continuous-time systems,” *IEEE Transactions on Automatic Control*, 2023.
- [5] P. Rapisarda and Y. Zhang, “An input-output ‘Fundamental Lemma’ for quarter-plane causal 2D models,” *IEEE Control Systems Letters*, 2024.
- [6] G. Pan, R. Ou, and T. Faulwasser, “On a stochastic fundamental lemma and its use for data-driven optimal control,” *IEEE Transactions on Automatic Control*, vol. 68, no. 10, pp. 5922–5937, 2022.
- [7] A. Padoan, F. Dörfler, and J. Lygeros, “Data-driven representations of conical, convex, and affine behaviors,” in *Proc. of the 62nd IEEE Conference on Decision and Control*, 2023, pp. 596–601.
- [8] B. Nortmann and T. Mylvaganam, “Direct data-driven control of linear time-varying systems,” *IEEE Transactions on Automatic Control*, vol. 68, no. 8, pp. 4888–4895, 2023.
- [9] M. Alsatti, V. G. Lopez, J. Berberich, F. Allgöwer, and M. A. Müller, “Data-based control of feedback linearizable systems,” *IEEE Transactions on Automatic Control*, vol. 68, no. 11, pp. 7014–7021, 2023.
- [10] A. Fazzi and A. Chiuso, “Data-driven prediction and control for narx systems,” *arXiv preprint arXiv:2304.02930*, 2023.
- [11] J. G. Rueda-Escobedo and J. Schiffer, “Data-driven internal model control of second-order discrete volterra systems,” in *Proc. of the 59th IEEE Conference on Decision and Control*, 2020, pp. 4572–4579.
- [12] R. Tóth and C. Verhoek, “Modeling and control of LPV systems,” in *Reference Module in Materials Science and Materials Engineering*, ser. Encyclopedia of Systems and Control Engineering. Elsevier, 2025, pp. 1–14.
- [13] W. J. Rugh and J. S. Shamma, “Research on gain scheduling,” *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.
- [14] R. Tóth, *Modeling and Identification of Linear Parameter-Varying Systems*, 1st ed. Springer-Verlag, 2010.
- [15] C. Hoffmann and H. Werner, “A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 416–433, 2014.
- [16] 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, 2005.
- [17] F. G. Nogueira, W. B. Junior, C. T. da Costa Junior, and J. J. Lana, “LPV-based power system stabilizer: Identification, control and field tests,” *Control Engineering Practice*, vol. 72, pp. 53–67, 2018.
- [18] S. Kwon, T. Shimomura, and H. Okubo, “Pointing control of spacecraft using two SGCMGs via LPV control theory,” *Acta Astronautica*, vol. 68, no. 7–8, pp. 1168–1175, 2011.
- [19] P. J. W. Koelwijin, “Analysis and control of nonlinear systems with stability and performance guarantees: A linear parameter-varying approach,” Ph.D. dissertation, Eindhoven University of Technology, 2023.
- [20] C. Verhoek and R. Tóth, “Kernel-based multi-step predictors for data-driven analysis and control of nonlinear systems through the velocity form,” Eindhoven University of Technology, Tech. Rep., 2024.
- [21] C. Verhoek, R. Tóth, S. Haesaert, and A. Koch, “Fundamental Lemma for Data-Driven Analysis of Linear Parameter-Varying Systems,” in *Proc. of the 60th IEEE Conference on Decision and Control*, 2021, pp. 5040–5046.
- [22] C. Verhoek, H. S. Abbas, R. Tóth, and S. Haesaert, “Data-driven predictive control for linear parameter-varying systems,” in *Proc. of the 4th IFAC Workshop on Linear Parameter-Varying Systems*, vol. 54, no. 8, 2021, pp. 101–108.
- [23] C. Verhoek, J. Berberich, S. Haesaert, R. Tóth, and H. S. Abbas, “A linear parameter-varying approach to data predictive control,” *arXiv preprint arXiv:2311.07140*, 2024.
- [24] C. Verhoek, R. Tóth, and H. S. Abbas, “Direct Data-Driven State-Feedback Control of Linear Parameter-Varying Systems,” *International Journal of Robust and Nonlinear Control*, pp. 1–23, 2024.
- [25] C. Verhoek, J. Berberich, S. Haesaert, F. Allgöwer, and R. Tóth, “Data-driven dissipativity analysis of linear parameter-varying systems,” *IEEE Transactions on Automatic Control*, vol. 69, no. 12, pp. 8603–8616, 2024.
- [26] I. Markovsky, C. Verhoek, and R. Tóth, “The most powerful unfalsified linear parameter-varying model with shifted-affine scheduling dependence,” *Submitted*, 2024.
- [27] J. W. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer, 1997, vol. 26.
- [28] J. C. Willems, “From time series to linear system: Part I. Finite dimensional linear time invariant systems,” *Automatica*, vol. 22, no. 5, pp. 561–580, 1986.
- [29] R. Tóth, H. S. Abbas, and H. Werner, “On the state-space realization of LPV input-output models: Practical approaches,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 1, pp. 139–153, 2011.
- [30] P. den Boef, P. B. Cox, and R. Tóth, “LPVCORE: MATLAB toolbox for LPV modelling, identification and control,” in *Proc. of the 19th IFAC Symposium on System Identification*, 2021, pp. 385–390.
- [31] M. Baştug, M. Petreczky, R. Tóth, R. Wisniewski, J. Leth, and D. Efimov, “Moment Matching Based Model Reduction for LPV State-Space Models,” in *Proc. of the 54th IEEE Conference on Decision and Control*. IEEE, 2015.
- [32] M. Petreczky, R. Tóth, and G. Mercère, “Realization theory for LPV state-space representations with affine dependence,” *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4667–4674, 2016.
- [33] I. Markovsky and P. Rapisarda, “Data-Driven Simulation and Control,” *International Journal of Control*, vol. 81, no. 12, p. 1946–1959, 2008.
- [34] P. Cisneros and H. Werner, “Nonlinear model predictive control for models in quasi-linear parameter varying form,” *International Journal of Robust & Nonlinear Control*, vol. 30, no. 10, pp. 3945–3959, 2020.
- [35] C. Hespe and H. Werner, “Convergence properties of fast quasi-LPV model predictive control,” in *Proc. of the 60th Conference on Decision and Control*, 2021, pp. 3869–3874.

APPENDIX I PROOF OF LEMMA 1

Before we can give the proof for Lemma 1, we need some intermediate results. We first write the LPV-SS representation (9) with (11) in a structured form. This, in turn, allows us to draw conclusions on the observability of (9) with (11). Finally, these are used to give the proof of Lemma 1.

A. Structured LPV-SS form

Through the state construction (12) and the affine parametrization of the scheduling dependent polynomial coefficients of the kernel/IO representations, we can write the LPV-SS representation in terms of $x(k)$, $u(k)$, $p(k) \otimes y(k)$, $p(k) \otimes u(k)$, and $p(k) \otimes p(k) \otimes u(k)$ by splitting up (11) into scheduling-independent and scheduling-dependent parts:

$$\begin{aligned} x(k+1) = & A_0 x(k) + B_0 u(k) + A_p p(k) \otimes x_1(k) \\ & + B_p p(k) \otimes u(k) + B_{pp} p(k) \otimes p(k) \otimes u(k), \end{aligned} \quad (39a)$$

$$y(k) = C x(k) + D_0 u(k) + D_p p(k) \otimes u(k), \quad (39b)$$

with A_0, \dots, D_p as in (40) on the next page. Note that $x_1(k)$ is as in (12b), i.e., $x_1(k) \in \mathbb{R}^{n_y}$. Formulation (39) separates the LTI behavior from the parameter-varying behavior. By substituting x_1 in $p(k) \otimes x_1(k)$ in (39a) with (12b), the state

$$A_0 = \begin{bmatrix} -a_{1,0} & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -a_{n_a-1,0} & 0 & \dots & I \\ -a_{n_a,0} & 0 & \dots & 0 \end{bmatrix}, \quad B_0 = \begin{bmatrix} b_{1,0}-a_{1,0}b_{0,0} \\ \vdots \\ b_{n_b-1,0}-a_{n_b-1,0}b_{0,0} \\ b_{n_b,0}-a_{n_b,0}b_{0,0} \end{bmatrix}, \quad B_{\text{PP}} = \begin{bmatrix} -a_{1,1}b_{0,1} & \dots & -a_{1,n_p}b_{0,n_p} \\ \vdots & \ddots & \vdots \\ -a_{n_a,1}b_{0,1} & \dots & -a_{n_a,n_p}b_{0,n_p} \end{bmatrix}, \quad D_0 = b_{0,0}, \quad (40a)$$

$$A_p = \begin{bmatrix} -a_{1,1} & \dots & -a_{1,n_p} \\ \vdots & \ddots & \vdots \\ -a_{n_a,1} & \dots & -a_{n_a,n_p} \end{bmatrix}, \quad B_p = \begin{bmatrix} b_{1,1} & \dots & b_{1,n_p} \\ \vdots & \ddots & \vdots \\ b_{n_b,1} & \dots & b_{n_b,n_p} \end{bmatrix}, \quad C = [I \ 0 \ \dots \ 0], \quad D_p = [b_{0,1} \ \dots \ b_{0,n_p}] \quad (40b)$$

equation is written as

$$\begin{aligned} x(k+1) = & A_0x(k) + B_0u(k) + A_p p(k) \otimes y(k) \\ & + \tilde{B}_p p(k) \otimes u(k) + \tilde{B}_{\text{PP}} p(k) \otimes p(k) \otimes u(k), \end{aligned} \quad (41)$$

where $\tilde{B}_p = B_p - A_p(I_{n_p} \otimes D_0)$ and $\tilde{B}_{\text{PP}} = B_{\text{PP}} - A_p(I_{n_p} \otimes D_p)$. With this form, we can express a length L trajectory in $\mathfrak{B}|_{[1,L]}$ as follows

$$\begin{aligned} \text{vec}(y|_{[1,L]}) = & \mathcal{O}_L \mathbf{x} + \mathcal{T}_L \text{vec}(u|_{[1,L]}) + \mathcal{O}_L^p \text{vec}(y|_{[1,L]}^p) \\ & + \mathcal{T}_L^p \text{vec}(u|_{[1,L]}^p) + \mathcal{T}_L^{\text{PP}} \text{vec}(u|_{[1,L]}^{\text{PP}}), \end{aligned} \quad (42a)$$

with $u|_{[1,L]}^{\text{PP}} = (p(k) \otimes p(k) \otimes u(k))_{k=1}^L$ and initial condition $\mathbf{x} \in \mathbb{R}^{n_x}$, where

$$\begin{aligned} \mathcal{O}_L = & \begin{bmatrix} C \\ CA_0 \\ \vdots \\ CA_0^{L-1} \end{bmatrix}, \quad \mathcal{T}_L = \begin{bmatrix} D_0 & 0 & \dots & 0 \\ CB_0 & D_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA_0^{L-1}B_0 & \dots & CB_0 & D_0 \end{bmatrix}, \\ \mathcal{O}_L^p = & \begin{bmatrix} 0 & 0 & \dots & 0 \\ CA_p & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA_0^{L-1}A_p & \dots & CA_p & 0 \end{bmatrix}, \\ \mathcal{T}_L^p = & \begin{bmatrix} D_p & 0 & \dots & 0 \\ C\tilde{B}_p & D_p & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA_0^{L-1}\tilde{B}_p & \dots & C\tilde{B}_p & D_p \end{bmatrix}, \\ \mathcal{T}_L^{\text{PP}} = & \begin{bmatrix} 0 & 0 & \dots & 0 \\ C\tilde{B}_{\text{PP}} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ CA_0^{L-1}\tilde{B}_{\text{PP}} & \dots & C\tilde{B}_{\text{PP}} & 0 \end{bmatrix}. \end{aligned} \quad (42b)$$

With these formulations, we now provide observability and minimality properties of the LPV-SS representation, which are required for the proof of Lemma 1.

B. Observability of the structured LPV-SS form

We first introduce the notion of *complete state-observability* from [14]:

Definition 1 (Complete state-observability [14]). An LPV-SS representation (9) is called *completely state-observable*, if for all $(u, x, y, p) \in \mathfrak{B}^{\text{SS}}$, $(u, x', y, p) \in \mathfrak{B}^{\text{SS}}$ it holds that $x = x'$.

From which, the next result follows:

Lemma 3. Given a shifted-affine LPV-IO realization (8) for which the polynomials in (7) are left-coprime. The LPV-SS representation with matrices (11) that is constructed from this LPV-IO representation is completely state-observable.

Proof. Consider (42) for some $L \geq n_x$. Computing \mathcal{O}_L for this L gives

$$\mathcal{O}_L = \begin{bmatrix} I & 0 & \dots & \dots & \dots & 0 \\ -a_{1,0} & I & 0 & \dots & \dots & 0 \\ a_{1,0}^2 - a_{2,0} & -a_{1,0} & I & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix},$$

i.e., \mathcal{O}_L is a tall, lower-triangular matrix with 1's on the diagonal, i.e., \mathcal{O}_L is full column-rank. Hence, for a given $(u|_{[1,L]}, y|_{[1,L]}, p|_{[1,L]}) \in \pi_{u,y,p} \mathfrak{B}^{\text{SS}}$, the initial state \mathbf{x} can be uniquely determined via (42), after which the state trajectory is governed by (41). This implies that the state trajectory for $k \geq 1$ is unique. Since this holds for every $L \geq n_x$, together with the time-invariance property of LPV systems, the state trajectory x corresponding to each $(u, y, p) \in \pi_{u,y,p} \mathfrak{B}^{\text{SS}}$ is unique, i.e., the representation is completely state-observable. ■

This result states that for any scheduling sequence in \mathfrak{B}_{P} , the representation (11) is observable. We have already established in Section III-A.2 that we can always obtain a minimal LPV-SS representation with static scheduling dependence for the considered behaviors. Note that this minimal realization will admit the same structure as discussed in Section I-A, i.e., we can always formulate (42) with an \mathbf{x} of dimension $\mathbf{n}(\mathfrak{B})$, where the matrices $\mathcal{O}_{T_r}, \dots, \mathcal{T}_{T_r}^{\text{PP}}$ are constructed as in (42b).

Finally, we have the following result, which is key in deriving the proof for Lemma 1. This result links the rank of the ‘LTI part’ of the observability matrix, i.e., \mathcal{O}_L to the invariant integer $\mathbf{L}(\mathfrak{B})$.

Proposition 1. Given a minimal (11) constructed from (8) where the polynomials in (7) are left-coprime. Then $\text{rank}(\mathcal{O}_L) = \mathbf{n}(\mathfrak{B})$, if and only if $L \geq \mathbf{L}(\mathfrak{B})$.

Proof. First note that by Lemma 3, the representation is observable for any $p \in \mathfrak{B}_{\text{P}}$, including $0 \in \mathfrak{B}_{\text{P}}$. Then it follows from [28, Thm. 6] and [14, Sec. 4.3] that the observability index and n_r , i.e., $\mathbf{L}(\mathfrak{B})$ are equal. This implies that $\text{rank}(\mathcal{O}_L) = \mathbf{n}(\mathfrak{B})$, if and only if $L \geq n_r = \mathbf{L}(\mathfrak{B})$, concluding the proof. ■

We are now ready to give the proof of Lemma 1.

C. Proof of Lemma 1

Proof. Take an arbitrary $p|_{[1,L]} \in \mathfrak{B}_{\text{P}}|_{[1,L]}$ and an associated $\text{col}(u|_{[1,L]}, y|_{[1,L]}) \in \mathfrak{B}_p|_{[1,L]}$. Now consider the formulation of this trajectory in terms of (42). We can write

$$\text{vec}(u|_{[1,L]}^p) = (p|_{[1,L]} \odot I_{n_u}) \text{vec}(u|_{[1,L]}),$$

similarly for $\text{vec}(y_{[1,L]}^p)$. Furthermore, note that $\text{vec}(u_{[1,L]}^{pp})$ can be written as

$$\text{vec}(u_{[1,L]}^{pp}) = (p_{[1,L]} \odot I_{n_u n_p}) \text{vec}(u_{[1,L]}^p).$$

Hence, with $\mathcal{P}^{nu} = p_{[1,L]} \odot I_{n_u}$, $\mathcal{P}^{ny} = p_{[1,L]} \odot I_{n_y}$, and $\mathcal{P}^{nun_p} = p_{[1,L]} \odot I_{n_u n_p}$, we can rewrite (42) with $x \in \mathbb{R}^{n(\mathfrak{B})}$ as

$$(I - \mathcal{O}_L^p \mathcal{P}^{ny}) \text{vec}(y_{[1,L]}) = \mathcal{O}_L x + (\mathcal{T}_L + \mathcal{T}_L^p \mathcal{P}^{nu} + \mathcal{T}_L^{pp} \mathcal{P}^{nun_p} \mathcal{P}^{nu}) \text{vec}(u_{[1,L]}). \quad (43)$$

Hence, to characterize the manifest behavior of the LPV system for a given scheduling sequence $p_{[1,L]}$, we can use (43) to express the dynamic relation of any trajectory in $\mathfrak{B}_p|_{[1,L]}$ with

$$\begin{bmatrix} I & 0 \\ 0 & I - \mathcal{O}_L^p \mathcal{P}^{ny} \end{bmatrix} \text{vec}(w_{[1,L]}) = \begin{bmatrix} 0 & I \\ \mathcal{O}_L & \mathcal{T}_L + \mathcal{T}_L^p \mathcal{P}^{nu} + \mathcal{T}_L^{pp} \mathcal{P}^{nun_p} \mathcal{P}^{nu} \end{bmatrix} \begin{bmatrix} x \\ \text{vec}(u_{[1,L]}) \end{bmatrix}. \quad (44)$$

As \mathcal{O}_L^p is a strictly lower block-triangular matrix and \mathcal{P}^{ny} is a block diagonal matrix, $\mathcal{O}_L^p \mathcal{P}^{ny}$ is always strictly lower triangular. Therefore,

$$\begin{bmatrix} I & 0 \\ 0 & I - \mathcal{O}_L^p \mathcal{P}^{ny} \end{bmatrix} \quad (45)$$

is nonsingular. Thus, we can write

$$\text{vec}(w_{[1,L]}) = \underbrace{\begin{bmatrix} I & 0 \\ 0 & I - \mathcal{O}_L^p \mathcal{P}^{ny} \end{bmatrix}}_{\mathfrak{B}_p}^{-1} \begin{bmatrix} 0 & I \\ \mathcal{O}_L & \mathcal{Q} \end{bmatrix} \begin{bmatrix} x \\ \text{vec}(u_{[1,L]}) \end{bmatrix}, \quad (46)$$

with $\mathcal{Q} = \mathcal{T}_L + \mathcal{T}_L^p \mathcal{P}^{nu} + \mathcal{T}_L^{pp} \mathcal{P}^{nun_p} \mathcal{P}^{nu}$. Since w, p were chosen arbitrarily, we conclude that the columns of \mathfrak{B}_p form a basis for $\mathfrak{B}_p|_{[1,L]}$. The dimension of this basis is purely governed by $\begin{bmatrix} 0 & I \\ \mathcal{O}_L & \mathcal{Q} \end{bmatrix}$, because (45) is nonsingular. To conclude the proof, observe that the LPV-SS representation is minimal and $L \geq \mathbf{L}(\mathfrak{B})$. Hence, Proposition 1 gives that

$$\text{rank} \left(\begin{bmatrix} 0 & I \\ \mathcal{O}_L & \mathcal{Q} \end{bmatrix} \right) = n(\mathfrak{B}) + m(\mathfrak{B})L,$$

i.e., $\text{rank}(\mathfrak{B}_p) = n(\mathfrak{B}) + m(\mathfrak{B})L = \dim(\text{image}(\mathfrak{B}_p)) = \dim(\mathfrak{B}_p|_{[1,L]})$. ■

APPENDIX II PROOF OF THEOREM 1

Proof. We first prove Item (i) \Leftrightarrow Item (ii). First, note that $\mathfrak{B}_p|_{[1,L]}$ is a linear subspace. From Lemma 1, we know that for $L \geq \mathbf{L}(\mathfrak{B})$, $\dim(\mathfrak{B}_p|_{[1,L]}) = n(\mathfrak{B}) + m(\mathfrak{B})L$. Hence, if (28) holds, then $\dim(\mathfrak{B}_p|_{[1,L]}) = \dim(\text{image}(\mathcal{H}_L(\check{w}_{[1,N_d]}) \mathcal{N}_p)) = n(\mathfrak{B}) + m(\mathfrak{B})L$, i.e., (29) holds, which concludes the '(28) \Rightarrow (29)' direction.

Now we show the '(28) \Leftarrow (29)' direction. Consider (25) and an arbitrary scheduling sequence $\hat{p}_{[1,L]} \in \mathfrak{B}_p|_{[1,L]}$. For any g that is both in the row space of $\mathcal{H}_L(\check{w}_{[1,N_d]})$ and in the kernel of $\mathcal{H}_L(\check{w}_{[1,N_d]}^p) - \hat{\mathcal{P}}^{nu} \mathcal{H}_L(\check{w}_{[1,N_d]})$, we obtain

a trajectory $\hat{w}_{[1,L]}$. Based on the derivations in Section IV-A, we conclude that $(\hat{w}_{[1,L]}, \hat{p}_{[1,L]})$ trivially satisfies (1), i.e., $(\hat{w}_{[1,L]}, \hat{p}_{[1,L]}) \in \mathfrak{B}|_{[1,L]}$ and thus $\hat{w}_{[1,L]} \in \mathfrak{B}_{\hat{p}}|_{[1,L]}$. Hence, for this fixed scheduling sequence $\hat{p}_{[1,L]}$, the space spanned by $\text{image}(\mathcal{H}_L(\check{w}_{[1,N_d]}) \mathcal{N}_{\hat{p}})$ is a subspace of $\mathfrak{B}_{\hat{p}}|_{[1,L]}$, i.e.,

$$\text{image}(\mathcal{H}_L(\check{w}_{[1,N_d]}) \mathcal{N}_{\hat{p}}) \subseteq \mathfrak{B}_{\hat{p}}|_{[1,L]}. \quad (47)$$

Therefore, as the dimension of $\mathfrak{B}_{\hat{p}}|_{[1,L]}$ is equal to $n(\mathfrak{B}) + m(\mathfrak{B})L$, if (29) holds, then (47) must hold with equality, i.e., (28) must hold. The proof of Item (iii) follows directly from the above reasoning and the derivations in Section IV-A. To prove Item (iv), note that through (19), the right-hand side of condition (30) is equivalent to the right-hand side of (21). Then, the proof follows from the fact that (21) \Leftrightarrow (20), from which (25) is obtained without any loss of equivalence. Since Item (iii) \Leftrightarrow Item (ii), we have that (25) \Leftrightarrow (29), which concludes the proof. ■

APPENDIX III PROOF OF LEMMA 2

Proof. Consider the direct LPV-SS realization of Σ with state dimension $n(\mathfrak{B})$ and behavior \mathfrak{B}^{ss} . Consider a trajectory $(u_{[k_1,k_2]}, y_{[k_1,k_2]}, p_{[k_1,k_2]}, x_{[k_1,k_2]}) \in \mathfrak{B}^{ss}|_{[k_1,k_2]}$, with $k_1 \geq k_2$, and let us denote $\bar{k} = k_2 - k_1$ and $y_{\bar{k}} := \text{vec}(y_{[k_1,k_2]})$ for brevity (similarly for u and p). From the discussion in Section I-A, we know that we can express $(u_{[k_1,k_2]}, y_{[k_1,k_2]}, p_{[k_1,k_2]}) \in \pi_{u,y,p} \mathfrak{B}^{ss}|_{[k_1,k_2]}$ in the form of (42):

$$(I - \mathcal{O}_{\bar{k}}^p \mathcal{P}_{[k_1,k_2]}^{ny}) y_{\bar{k}} = \mathcal{O}_{\bar{k}} x(k_1) + \mathcal{Q}_{\bar{k}} u_{\bar{k}} \quad (48)$$

where

$$\mathcal{Q}_{\bar{k}} = \mathcal{T}_{\bar{k}} + \mathcal{T}_{\bar{k}}^p \mathcal{P}_{[k_1,k_2]}^{nu} + \mathcal{T}_{\bar{k}}^{pp} \mathcal{P}_{[k_1,k_2]}^{nun_p} \mathcal{P}_{[k_1,k_2]}^{nu},$$

with $\mathcal{P}_{[k_1,k_2]}^{nu} = p_{[k_1,k_2]} \odot I_{n_u}$, $\mathcal{P}_{[k_1,k_2]}^{ny} = p_{[k_1,k_2]} \odot I_{n_y}$, and $\mathcal{P}_{[k_1,k_2]}^{nun_p} = p_{[k_1,k_2]} \odot I_{n_u n_p}$. Furthermore, we can also express $x(k_2)$ by recursive application of (41) starting from $x(k_1)$:

$$x(k_2) = A_0^{k_2 - k_1} x(k_1) + \mathcal{A}_{\bar{k}} \mathcal{P}_{[k_1,k_2]}^{ny} y_{\bar{k}} + \mathcal{B}_{\bar{k}} u_{\bar{k}} + \mathcal{B}_{\bar{k}}^p \mathcal{P}_{[k_1,k_2]}^{nu} u_{\bar{k}} + \mathcal{B}_{\bar{k}}^{pp} \mathcal{P}_{[k_1,k_2]}^{nun_p} \mathcal{P}_{[k_1,k_2]}^{nu} u_{\bar{k}}, \quad (49)$$

where

$$\begin{aligned} \mathcal{A}_{\bar{k}} &= [A_0^{\bar{k}-1} A_p \dots A_p 0], & \mathcal{B}_{\bar{k}} &= [A_0^{\bar{k}-1} B_0 \dots B_0 0], \\ \mathcal{B}_{\bar{k}}^p &= [A_0^{\bar{k}-1} \tilde{B}_p \dots \tilde{B}_p 0], & \mathcal{B}_{\bar{k}}^{pp} &= [A_0^{\bar{k}-1} \tilde{B}_{pp} \dots \tilde{B}_{pp} 0]. \end{aligned}$$

We will now express $x(k_2)$ in terms of only the trajectories $(\text{col}(u_{[k_1,k_2]}, y_{[k_1,k_2]}), p_{[k_1,k_2]})$. Suppose $\mathcal{O}_{\bar{k}}$ has a left-inverse $\mathcal{O}_{\bar{k}}^+$, then we obtain the following expression for $x(k_1)$:

$$x(k_1) = \mathcal{O}_{\bar{k}}^+ (I - \mathcal{O}_{\bar{k}}^p \mathcal{P}_{[k_1,k_2]}^{ny}) y_{\bar{k}} - \mathcal{O}_{\bar{k}}^+ \mathcal{Q}_{\bar{k}} u_{\bar{k}}. \quad (50)$$

Substitution of (50) in (49) gives

$$\begin{aligned} x(k_2) &= \left(A_0^{k_2 - k_1} \mathcal{O}_{\bar{k}}^+ (I - \mathcal{O}_{\bar{k}}^p \mathcal{P}_{[k_1,k_2]}^{ny}) + \mathcal{A}_{\bar{k}} \mathcal{P}_{[k_1,k_2]}^{ny} \right) y_{\bar{k}} \\ &\quad + \left(A_0^{k_2 - k_1} \mathcal{O}_{\bar{k}}^+ \mathcal{Q}_{\bar{k}} + \mathcal{B}_{\bar{k}} + \mathcal{B}_{\bar{k}}^p \mathcal{P}_{[k_1,k_2]}^{nu} \dots \right. \\ &\quad \left. + \mathcal{B}_{\bar{k}}^{pp} \mathcal{P}_{[k_1,k_2]}^{nun_p} \mathcal{P}_{[k_1,k_2]}^{nu} \right) u_{\bar{k}}, \end{aligned} \quad (51)$$

i.e., $x(k_2)$ can be uniquely expressed from the trajectory $(\text{col}(u_{[k_1, k_2]}, y_{[k_1, k_2]}), p_{[k_1, k_2]})$ given that $\mathcal{O}_{\bar{k}}$ has a left-inverse. From Proposition 1, we have that $\mathcal{O}_{\bar{k}}$ is full column rank, i.e., $\mathcal{O}_{\bar{k}}$ has a left-inverse, if and only if $\bar{k} \geq \mathbf{L}(\mathfrak{B})$. For $k_1 = 1$ and $k_2 = T_i + 1$, $x(k_2)$ is the initial condition for the trajectory (w_r, p_r) , which can be uniquely expressed in terms of (w_i, p_i) and⁷ $w(T_i + 1), p(T_i + 1)$ if and only if $\bar{k} = T_i \geq \mathbf{L}(\mathfrak{B})$, concluding the proof. ■



Chris Verhoeck received his M.Sc. degree (Cum Laude) in Systems and Control from the Eindhoven University of Technology (TU/e) in 2020. In 2025, he received his Ph.D. degree with Cum Laude distinction, also from the TU/e. His M.Sc. thesis was selected as best thesis of the Electrical Engineering department in the year 2020. He is currently a postdoctoral researcher with the Control Systems group, Dept. of Electrical Engineering, TU/e. In the fall of 2023, he was a visiting researcher at the IfA, ETH Zürich, Switzerland. His main research interests include (data-driven) analysis and control of nonlinear and LPV systems and learning-for-control techniques with stability and performance guarantees.



Roland Tóth received his Ph.D. degree with Cum Laude distinction at the Delft University of Technology (TUDelft) in 2008. He was a postdoctoral researcher at TUDelft in 2009 and at the Berkeley Center for Control and Identification, University of California in 2010. He held a position at TUDelft in 2011-12, then he joined to the Control Systems (CS) Group at the Eindhoven University of Technology (TU/e). Currently, he is a Full Professor at the CS Group, TU/e and a Senior Researcher at the Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control (SZTAKI) in Budapest, Hungary. He is Senior Editor of the IEEE Transactions on Control Systems Technology and Associate Editor of Automatica. His research interests are in identification and control of linear parameter-varying (LPV) and nonlinear systems, developing data-driven and machine learning methods with performance and stability guarantees for modeling and control, model predictive control and behavioral system theory. On the application side, his research focuses on advancing reliability and performance of precision mechatronics and autonomous robots/vehicles with nonlinear, LPV and learning-based motion control. He has received the TUDelft Young Researcher Fellowship Award in 2010, the VENI award of The Netherlands Organization for Scientific Research in 2011, the Starting Grant of the European Research Council in 2016 and the DCRG Fellowship of Mathworks in 2022.



Ivan Markovsky received the Ph.D. degree in electrical engineering from the Katholieke Universiteit Leuven, Leuven, Belgium, in February 2005. He is currently an ICREA Professor with the International Centre for Numerical Methods in Engineering, Barcelona. From 2006 to 2012, he was an Assistant Professor with the School of Electronics and Computer Science, University of Southampton, Southampton, U.K., and from 2012 to 2022, an Associate Professor with the Vrije Universiteit, Brussel, Belgium. His research interests are computational methods for system theory, identification, and data-driven control in the behavioral setting. Dr. Markovsky was the recipient of an ERC starting grant "Structured low-rank approximation: Theory, algorithms, and applications" 2010-2015, Householder Prize honorable mention 2008, and research mandate by the Vrije Universiteit Brussel research council 2012-2022.



Sofie Haesaert received the B.Sc. degree cum laude in mechanical engineering and the M.Sc. degree cum laude in systems and control from the Delft University of Technology, Delft, The Netherlands, in 2010 and 2012, respectively, and the Ph.D. degree from Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands, in 2017. She is currently an Associate Professor with the Control Systems Group, Department of Electrical Engineering, TU/e. From 2017 to 2018, she was a Postdoctoral Scholar with Caltech. Her research interests are in the identification, verification, and control of cyber-physical systems for temporal logic specifications and performance objectives.

⁷Note that if there is no feed-through, i.e., $D(p) = b_0(p) = 0$, then (51) can be expressed in terms of only $(w_{[k_1, k_2-1]}, p_{[k_1, k_2-1]})$.