



HAL
open science

Realization and identification algorithm for stochastic LPV state-space models with exogenous inputs

Manas Mejari, Mihaly Petreczky

► **To cite this version:**

Manas Mejari, Mihaly Petreczky. Realization and identification algorithm for stochastic LPV state-space models with exogenous inputs. 3rd IFAC Workshop on Linear Parameter-Varying Systems, Nov 2019, Eindhoven, Netherlands. 10.1016/j.ifacol.2019.12.340 . hal-02398576

HAL Id: hal-02398576

<https://hal.science/hal-02398576>

Submitted on 31 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Realization and identification algorithm for stochastic LPV state-space models with exogenous inputs [★]

Manas Mejari ^{*} Mihály Petreczky ^{**}

^{*} *IDSIA Dalle Molle Institute for Artificial Intelligence SUPSI, 6928 Manno, Switzerland (e-mail: manas.mejari@idsia.ch)*

^{**} *Centre de Recherche en Informatique, Signal et Automatique de Lille, UMR CNRS 9189, Ecole Centrale de Lille, Villeneuve d'Ascq 59651, France (mihaly.petreczky@ec-lille.fr)*

Abstract: In this paper, we present a realization and an identification algorithm for stochastic *Linear Parameter-Varying State-Space Affine* (LPV-SSA) representations. The proposed realization algorithm combines the deterministic LPV input output to LPV state-space realization scheme based on correlation analysis with a stochastic covariance realization algorithm. Based on this realization algorithm, a computationally efficient and statistically consistent identification algorithm is proposed to estimate the LPV model matrices, which are computed from the empirical covariance matrices of outputs, inputs and scheduling signal observations. The effectiveness of the proposed algorithm is shown via a numerical case study.

1. INTRODUCTION

Identification of *Linear Parameter-Varying* (LPV) models has gained significant attention over the past few years, owing to their ability to describe the behavior of many time-varying and non-linear systems. Many approaches have been proposed for the identification of LPV models, in input-output (Bamieh and Giarré, 2002; Laurain et al., 2010; Mejari et al., 2018; Piga et al., 2015) as well as State-Space (SS) representations (Felici et al., 2007; Tanelli et al., 2011; van Wingerden and Verhaegen, 2009; Verdult and Verhaegen, 2005). The reader is referred to (Tóth, 2010) for a detailed summary of the available LPV identification approaches.

Controller design approaches often require the LPV models to be in SS representation with an affine dependency on the scheduling variable. To this end, *realization theory* of LPV models plays a key role in understanding the conditions under which the observed behavior of a system can be realized by a state-space affine representation. It also allows to formulate identification algorithms for estimating state-space representation from a finite set of observations. The realization theory for deterministic *Linear Parameter-Varying State-Space with Affine dependence* (LPV-SSA) representation has been developed in Tóth et al. (2012); Petreczky et al. (2017). The results of Tóth et al. (2012); Petreczky et al. (2017) were used to derive LPV-SS identification algorithm in Cox et al. (2015, 2018). These methods are focused on *deterministic* realizations, which for certain control and filtering problems are too restrictive. In this paper,

we focus on formulating a realization algorithm and a related identification algorithm for *stochastic* LPV-SSA representations. The main idea is to decompose the stochastic LPV-SSA realization/identification problem into two independent problems: realization/identification of *deterministic* part which depends only on the input, and realization/identification of *stochastic* part. To this end, the proposed algorithm is based on the combination of correlation analysis (Cox et al., 2018) for deterministic realization and stochastic covariance identification algorithm for stochastic LPV-SSA representations (Mejari and Petreczky, 2019a).

The algorithm presented in this paper extends the results of Mejari and Petreczky (2019a), to the case of stochastic LPV-SSA representations with exogenous inputs. The proposed approach differs significantly from the subspace based identification methods for stochastic LPV-SSA representations (van Wingerden and Verhaegen, 2009; dos Santos et al., 2009; Favoreel et al., 1999). First, the cited papers do not deal with the realization problem. In particular, while the possibility of decomposing the output into a deterministic and purely stochastic components is sometimes claimed in the literature, the formal details of such a decomposition were never addressed. Second, in contrast to the literature mentioned above, the identification algorithm proposed in this paper is provenly consistent and it does not require local observability assumptions. The downside is that the proposed algorithm is provenly consistent only for a specific class of scheduling signals and stochastic LPV-SSA representations. Moreover, the proposed algorithm avoids the curse of dimensionality, but this comes at a price of either using some prior knowledge on the system to determine the correct selection of the rows and columns of a Hankel-matrix or using an exhaustive search to find such a selection.

[★] This work was partially funded by CPER Data project, co-financed by European Union with the financial support of European Regional Development Fund, French State and the French Region of Hauts-de-France, and by CNRS project PEPS blanc BayesRealForRNN.

The paper is organized as follows. In Section 2, we present the problem formulation. Section 3 presents the formal definition and basic properties of the class of LPV state-space representations considered in this paper. In Section 4, we formalize the decomposition of outputs of such LPV state-space representations into stochastic and deterministic components. In Section 5, we present the realization algorithm for stochastic LPV state-space representations, and in Section 6 we present the related identification algorithm. Finally, in Section 7 we illustrate the results with a numerical example.

2. PROBLEM FORMULATION

Let \mathbf{y} , \mathbf{u} , $\boldsymbol{\mu}$ be stochastic processes taking values in \mathbb{R}^{n_y} , \mathbb{R}^{n_u} and \mathbb{R}^{n_μ} , representing the output process, the input process, and the scheduling signal process respectively. We define a discrete-time *Linear Parameter-Varying State-Space Affine* (LPV-SSA) representation of the process $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$ as the discrete-time system of the form

$$\begin{aligned} \mathbf{x}(t+1) &= \sum_{i=1}^{n_\mu} (A_i \mathbf{x}(t) + B_i \mathbf{u}(t) + K_i \mathbf{v}(t)) \boldsymbol{\mu}_i(t), \\ \mathbf{y}(t) &= C \mathbf{x}(t) + D \mathbf{u}(t) + \mathbf{v}(t), \end{aligned} \quad (1)$$

where, $A_i \in \mathbb{R}^{n_x \times n_x}$, $B_i \in \mathbb{R}^{n_x \times n_u}$, $K_i \in \mathbb{R}^{n_x \times n_y}$, $\forall i = 1, \dots, n_\mu$, $C \in \mathbb{R}^{n_y \times n_x}$ and $D \in \mathbb{R}^{n_y \times n_u}$ are real constant matrices, and \mathbf{v} is a white noise process, i.e., $E[\mathbf{v}(t)\mathbf{v}^T(s)] = 0$, $s \neq t$ and $E[\mathbf{v}(t)\mathbf{v}^T(t)\boldsymbol{\mu}_i(t)] = Q_i > 0$, $i = 1, \dots, n_\mu$. The realization and identification problems considered in this paper are as follows.

Problem 1. (Realization problem). For process $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$, find matrices $(\{A_i, B_i, K_i\}_{i=1}^{n_\mu}, C, D)$ and processes \mathbf{x}, \mathbf{v} such that (1) is a representation of $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$.

Problem 2. (Identification problem). Assume that $y : \mathbb{Z} \rightarrow \mathbb{R}^{n_y}$ is a sample path of the output process \mathbf{y} , $u : \mathbb{Z} \rightarrow \mathbb{R}^{n_u}$ is a sample path of the input process \mathbf{u} and $\mu : \mathbb{Z} \rightarrow \mathbb{R}^{n_\mu}$ is a sample path of the scheduling process $\boldsymbol{\mu}$, corresponding to the same random event $\omega \in \Omega$. Given a dataset $\{y(t), u(t), \mu(t)\}_{t=1}^N$ consisting of N samples of the output, input and scheduling process, compute from this dataset the estimates $\{\{\hat{A}_i^N, \hat{B}_i^N, \hat{K}_i^N, \hat{Q}_i^N\}_{i=1}^{n_\mu}, \hat{C}^N, \hat{D}^N\}$, such that as $N \rightarrow \infty$, the estimated matrices $\{\{\hat{A}_i^N, \hat{B}_i^N, \hat{K}_i^N, \hat{Q}_i^N\}_{i=1}^{n_\mu}, \hat{C}^N, \hat{D}^N\}$ converge to matrices $\{A_i, B_i, K_i, Q_i\}_{i=1}^{n_\mu}, C, D$ such that the LPV-SSA (1) with $Q_i = E[\mathbf{v}(t)\mathbf{v}^T(t)\boldsymbol{\mu}_i^2(t)]$, $i = 1, \dots, n_\mu$, is a representation of $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$.

3. PROPERTIES OF LPV-SSA REPRESENTATION

In order to make Problems 1-2 well-posed, we have to impose additional constraints on the class of processes $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$ and on the class of LPV-SSA representations. To this end, we need the following notation and terminology.

Notation 1. (Σ). Let $\Sigma = \{1, \dots, n_\mu\}$.

A *non empty word* over Σ is a finite sequence of letters, i.e., $w = \sigma_1 \sigma_2 \dots \sigma_k$, where $0 < k \in \mathbb{Z}$, $\sigma_1, \sigma_2, \dots, \sigma_k \in \Sigma$. The set of *all* nonempty words is denoted by Σ^+ . We denote an *empty word* by ϵ . Let $\Sigma^* = \epsilon \cup \Sigma^+$. The concatenation of two nonempty words $v = a_1 a_2 \dots a_m$ and $w = b_1 b_2 \dots b_n$ is defined as $vw = a_1 \dots a_m b_1 \dots b_n$ for some $m, n > 0$. Note that if $w = \epsilon$ or $v = \epsilon$, then $v\epsilon = v$ and $\epsilon w = w$, moreover,

$\epsilon\epsilon = \epsilon$. The length of the word $w \in \Sigma^*$ is denoted by $|w|$, and $|\epsilon| = 0$.

Assumption 1. (White noise scheduling). The scheduling process $\boldsymbol{\mu} = [1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_{n_\mu}]^T$ is zero-mean independent identically distributed (i.i.d.) such that, for all $t \in \mathbb{Z}$, we have $\boldsymbol{\mu}_1(t) \equiv 1$, and for each $\sigma = 2, \dots, n_\mu$, $\boldsymbol{\mu}_\sigma$ is a zero mean i.i.d. process.

We define scalars $E[\boldsymbol{\mu}_\sigma^2(t)] = p_\sigma$, for all $t \in \mathbb{Z}$. In particular, $p_1 = 1$. For every word $w \in \Sigma^+$ where $w = \sigma_1 \sigma_2 \dots \sigma_k$, $k \geq 1$, $\sigma_1, \dots, \sigma_k \in \Sigma$, we define the process $\boldsymbol{\mu}_w$ and the number p_w as follows

$$\begin{aligned} \boldsymbol{\mu}_w(t) &= \boldsymbol{\mu}_{\sigma_1}(t - k + 1) \boldsymbol{\mu}_{\sigma_2}(t - k + 2) \dots \boldsymbol{\mu}_{\sigma_k}(t), \forall t \in \mathbb{Z} \\ p_w &= p_{\sigma_1} p_{\sigma_2} \dots p_{\sigma_k}. \end{aligned}$$

We set $\boldsymbol{\mu}_\epsilon(t) = 1$ and $p_\epsilon = 1$. For a process $\mathbf{r} \in \mathbb{R}^{n_u}$, for each $w \in \Sigma^+$ we define the process $\mathbf{z}_w^{\mathbf{r}}$ as

$$\mathbf{z}_w^{\mathbf{r}}(t) = \mathbf{r}(t - |w|) \boldsymbol{\mu}_w(t - 1) \frac{1}{\sqrt{p_w}}, \quad \forall t \in \mathbb{Z}, \quad (2)$$

which is interpreted as the *past* of \mathbf{r} w.r.t. $\{\boldsymbol{\mu}_\sigma\}_{\sigma \in \Sigma}$.

We remark that in this paper, the notion of *Zero Mean Wide Sense Stationary w.r.t. Inputs* (ZMWSSI) process and *Square Integrable w.r.t. Input* (SII) process, will be a central notion for the mathematical framework of stochastic LPV-SSA representations. We refer the reader to Petreczky and Vidal (2018) for the detailed definition of ZMWSSI and SII processes.

All the process considered in this paper will be assumed to be ZMWSSI and SII process w.r.t. scheduling $\boldsymbol{\mu}$.

Definition 1. (White noise w.r.t. $\boldsymbol{\mu}$). A process \mathbf{r} is called a white noise process w.r.t. $\boldsymbol{\mu}$, if \mathbf{r} is ZMWSII w.r.t. $\boldsymbol{\mu}$, and $E[\mathbf{z}_w^{\mathbf{r}}(t)(\mathbf{z}_v^{\mathbf{r}}(t))^T] = 0$, $v \neq w$, $E[\mathbf{z}_\sigma^{\mathbf{r}}(t)(\mathbf{z}_{\sigma w}^{\mathbf{r}}(t))^T] = E[\mathbf{z}_\sigma^{\mathbf{r}}(t)(\mathbf{z}_\sigma^{\mathbf{r}}(t))^T] > 0$, for all $w, v \in \Sigma^+$.

Using the concept of ZMWSSI process and white noise process w.r.t. $\boldsymbol{\mu}$, we can formulate the main assumption regarding the processes $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$.

Assumption 2. Assume that $\boldsymbol{\mu}$ satisfies Assumption 1, and $[\mathbf{y}^T \mathbf{u}^T]^T$ is a ZMWSSI and SII process w.r.t. $\boldsymbol{\mu}$, and \mathbf{u} is a white noise process w.r.t. $\boldsymbol{\mu}$, and the covariance $E[\mathbf{z}_\sigma^{\mathbf{u}}(t)(\mathbf{z}_\sigma^{\mathbf{u}}(t))^T] = E[\mathbf{u}(t-1)(\mathbf{u}(t-1))^T] = \Lambda_u > 0$ does not depend on $\sigma \in \Sigma$.

Next, we recall from Mejari and Petreczky (2019a) the notion of a *stationary* stochastic LPV-SSA representation of a process \mathbf{r} without inputs.

Definition 2. A *stationary LPV-SSA representation without inputs* of a process \mathbf{r} taking values in \mathbb{R}^p , is a tuple $(\{\tilde{A}_\sigma, \tilde{K}_\sigma\}_{\sigma=1}^{n_\mu}, \tilde{C}, \tilde{D}, \tilde{\mathbf{x}}, \tilde{\mathbf{v}})$, where $\tilde{A}_\sigma \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$, $\tilde{K}_\sigma \in \mathbb{R}^{\tilde{n} \times \tilde{m}}$, $\tilde{C} \in \mathbb{R}^{p \times \tilde{n}}$ and $\tilde{\mathbf{v}}$ is a process taking values in $\mathbb{R}^{\tilde{m}}$ such that such that

1. $[\tilde{\mathbf{x}}^T \tilde{\mathbf{v}}^T]^T$ is a ZMWSSI process, and $E[\tilde{\mathbf{x}}(t)(\mathbf{z}_w^{\tilde{\mathbf{x}}}(t))^T] = 0$ for all $w \in \Sigma^+$.
2. $\tilde{\mathbf{v}}$ is a white noise process w.r.t. $\boldsymbol{\mu}$.
3. The eigenvalues of the matrix $\sum_{\sigma \in \Sigma} p_\sigma \tilde{A}_\sigma \otimes \tilde{A}_\sigma$ are inside the open unit circle.
4. $\tilde{\mathbf{x}}(t+1) = \sum_{i=1}^{n_\mu} (\tilde{A}_i \tilde{\mathbf{x}}(t) + \tilde{K}_i \tilde{\mathbf{v}}(t)) \boldsymbol{\mu}_i(t)$, $\mathbf{r}(t) = \tilde{C} \tilde{\mathbf{x}}(t) + \tilde{D} \tilde{\mathbf{v}}(t)$.

We call $\tilde{\mathbf{x}}$ the state process and $\tilde{\mathbf{v}}$ the noise process.

In the terminology of Petreczky and Vidal (2018), a stationary LPV-SSA without inputs \mathbf{u} , corresponds to a stationary generalized bilinear system w.r.t. the scheduling inputs $\{\boldsymbol{\mu}_\sigma\}_{\sigma \in \Sigma}$. If a process \mathbf{r} has a stationary LPV-SSA representation without inputs, then \mathbf{r} is a ZMWSSI process and $\tilde{\mathbf{x}}$ is uniquely determined by $\tilde{\mathbf{v}}$ and the matrices $(\hat{C}, \hat{D}, \{\hat{A}_\sigma, \hat{K}_\sigma\}_{\sigma \in \Sigma})$. In order to define this notion more precisely, let us introduce the following notation.

Notation 2. (Matrix Product). Consider a collection of square matrices $A_\sigma \in \mathbb{R}^{n \times n}$, $\sigma \in \Sigma$. For any word $w \in \Sigma^+$ of the form $w = \sigma_1 \sigma_2 \cdots \sigma_k$, $k > 0$ and $\sigma_1, \dots, \sigma_k \in \Sigma$, we define $A_w = A_{\sigma_k} \cdots A_{\sigma_2} A_{\sigma_1}$. For an empty word ϵ , $A_\epsilon = I_n$.

From Petreczky and Vidal (2018) it follows that

$$\tilde{\mathbf{x}}(t) = \sum_{\sigma \in \Sigma, w \in \Sigma^*} \sqrt{p_{\sigma w}} \tilde{A}_w \tilde{K}_\sigma \tilde{\mathbf{z}}_{\sigma w}^{\tilde{\mathbf{v}}}(t), \quad (3)$$

where the infinite sum on the right-hand side is absolutely convergent in the mean square sense.

Using the notion of a stationary LPV-SSA without inputs, we can define the class of LPV-SSA representation *with inputs* which will be considered in this paper.

Definition 3. (Stationary LPV-SSA). The LPV-SSA representation (1) is *stationary with input* \mathbf{u} , if $(\{A_\sigma, [K_\sigma \ B_\sigma]\}_{\sigma \in \Sigma}, C, [I_{n_y} \ D] \mathbf{x}, [\mathbf{v}^T \ \mathbf{u}^T]^T)$ is a stationary LPV-SSA representation of \mathbf{y} without inputs as in Definition 2, and the orthogonality condition $E[\mathbf{v}(t) \mathbf{u}^T(t) \boldsymbol{\mu}_\sigma^2(t)] = 0$, $\forall \sigma \in \Sigma$ holds.

From (3) it follows that for a stationary LPV-SSA representation with input \mathbf{u} of the form (1),

$$\mathbf{x}(t) = \sum_{w \in \Sigma^*, \sigma \in \Sigma} \sqrt{p_{\sigma w}} A_w (K_\sigma \mathbf{z}_{\sigma w}^{\mathbf{v}}(t) + B_\sigma \mathbf{z}_{\sigma w}^{\mathbf{u}}(t)),$$

where the infinite sums on the right hand side are absolutely convergent in the mean-square sense. That is, the matrices and the noise processes determine the state process of a stationary LPV-SSA (with or without inputs) uniquely.

4. DECOMPOSITION OF THE OUTPUT OF LPV-SSA REPRESENTATION

It turns out that the output process of stationary LPV-SSA representations admits a decomposition into *deterministic* and *stochastic* parts. The deterministic part depends only on the input process, while the stochastic part depends only on the noise process. This decomposition does not depend on the particular choice of LPV-SSA representation, but only on the output process at hand.

Definition 4. (Deterministic and stochastic components). Assume the processes $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$ satisfy Assumption 2. Define the *deterministic component* \mathbf{y}^d of \mathbf{y} as follows

$$\mathbf{y}^d(t) = E_l[\mathbf{y} \mid \{\mathbf{z}_w^{\mathbf{u}}(t)\}_{w \in \Sigma^+} \cup \{\mathbf{u}(t)\}], \quad (4)$$

where $E_l[\cdot]$ is the *orthogonal projection* as defined in Petreczky and Vidal (2018). Define the *stochastic component* of \mathbf{y} as

$$\mathbf{y}^s(t) = \mathbf{y}(t) - \mathbf{y}^d(t). \quad (5)$$

From the definition it follows that

$$\mathbf{y}(t) = \mathbf{y}^d(t) + \mathbf{y}^s(t),$$

i.e., the process $\mathbf{y}(t)$ can be represented as the sum of its deterministic and stochastic components. In case when the process admits an LPV-SSA representation, the stochastic and deterministic components satisfy the following properties.

Lemma 1. (Decomposition of \mathbf{y}). Assume that there exists a stationary LPV-SSA representation of $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$ of the form (1) and that $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$ satisfy Assumption 2. It then follows that

$$\begin{aligned} \mathbf{x}^d(t+1) &= \sum_{i=1}^{n_\mu} (A_i \mathbf{x}^d(t) + B_i \mathbf{u}(t)) \boldsymbol{\mu}_i(t), \\ \mathbf{y}^d(t) &= C \mathbf{x}^d(t) + D \mathbf{u}(t), \end{aligned} \quad (6)$$

and $(\{A_\sigma, B_\sigma\}_{\sigma \in \Sigma}, C, D, \mathbf{x}^d, \mathbf{u})$ is a stationary LPV-SSA representation of \mathbf{y}^d without inputs and with noise process \mathbf{u} , moreover,

$$\begin{aligned} \mathbf{x}^s(t+1) &= \sum_{i=1}^{n_\mu} (A_i \mathbf{x}^s(t) + K_i \mathbf{v}(t)) \boldsymbol{\mu}_i(t), \\ \mathbf{y}^s(t) &= C \mathbf{x}^s(t) + \mathbf{v}(t), \end{aligned} \quad (7)$$

and $(\{A_\sigma, K_\sigma\}_{\sigma \in \Sigma}, C, I_{n_y}, \mathbf{x}^s, \mathbf{v})$ is a stationary LPV-SSA representation of \mathbf{y}^s without inputs, where

$$\mathbf{x}^d(t) = E_l[\mathbf{x}(t) \mid \{\mathbf{z}_w^{\mathbf{u}}(t)\}_{w \in \Sigma^+} \cup \{\mathbf{u}(t)\}] \quad (8)$$

$$\mathbf{x}^s(t) = \mathbf{x}(t) - \mathbf{x}^d(t) \quad (9)$$

The proof of Lemma 1 is presented in Mejari and Petreczky (2019b). Thus, \mathbf{y}^s depends only on the noise \mathbf{v} , and \mathbf{y}^d does not depend on the noise but it depends only on input \mathbf{u} . In fact, the converse of Lemma 1 also holds.

Lemma 2. Assume that \mathbf{y} has a stationary LPV-SSA representation with input \mathbf{u} . Assume that $\Sigma_d = (\{\hat{A}_i^d, \hat{B}_i^d\}_{i=1}^{n_\mu}, \hat{C}^d, \hat{D}^d, \hat{\mathbf{x}}^d, \mathbf{u})$ is a stationary LPV-SSA representation of \mathbf{y}^d without input such that its noise process equals the input process \mathbf{u} . Assume that $\Sigma_s = (\{\hat{A}_i^s, \hat{K}_i^s\}_{i=1}^{n_\mu}, \hat{C}^s, I_{n_y}, \hat{\mathbf{x}}^s, \mathbf{e}^s)$ is a stationary LPV-SSA representation of \mathbf{y}^s without inputs in *forward innovation form*, i.e., assume that the process \mathbf{e}^s is the so called *innovation process* of \mathbf{y}^s as defined in Petreczky and Vidal (2018):

$$\mathbf{e}^s(t) = \mathbf{y}^s(t) - E_l[\mathbf{y}^s(t) \mid \{\mathbf{z}_w^{\mathbf{y}^s}(t)\}_{w \in \Sigma^+}] \quad (10)$$

Then, tuple $(\{\hat{A}_i, \hat{K}_i, \hat{B}_i\}_{i=1}^{n_\mu}, \hat{C}, \hat{D}, \hat{\mathbf{x}}, \mathbf{e}^s)$ is a stationary LPV-SSA representation of \mathbf{y} with input \mathbf{u} , where

$$\begin{aligned} \hat{\mathbf{x}}(t) &= [(\hat{\mathbf{x}}^d(t))^T (\hat{\mathbf{x}}^s(t))^T]^T \\ \hat{A}_\sigma &= \text{diag}(\hat{A}_\sigma^d, \hat{A}_\sigma^s), \quad \hat{B}_\sigma = [(\hat{B}_\sigma^d)^T \ \mathbf{0}_{n_x \times n_u}^T]^T \\ \hat{K}_\sigma &= [\mathbf{0}_{n_x \times n_y}^T \ (\hat{K}_\sigma^s)^T]^T, \quad \hat{C} = [\hat{C}^d \ \hat{C}^s], \quad \hat{D} = \hat{D}^d. \end{aligned} \quad (11)$$

Moreover, the innovation process \mathbf{e}^s satisfies

$$\mathbf{e}^s(t) = \mathbf{y}(t) - E_l[\mathbf{y}(t) \mid \{\mathbf{z}_w^{\mathbf{y}}(t), \mathbf{z}_w^{\mathbf{u}}(t)\}_{w \in \Sigma^+} \cup \{\mathbf{u}(t)\}] \quad (12)$$

The proof of Lemma 2 is presented in Mejari and Petreczky (2019b). Thus, the problem of realization of \mathbf{y} can be decomposed into two problems:

- P1** finding a stationary LPV-SSA representation Σ_d without inputs of \mathbf{y}^d , such that the noise process of Σ_d is \mathbf{u} ,
- P2** finding a stationary LPV-SSA representation Σ_s without inputs of $\mathbf{y}^s = \mathbf{y} - \mathbf{y}^d$, such that the noise

process \mathbf{e}^s of Σ_s is the innovation process of \mathbf{y}^s as defined in Mejari and Petreczky (2019a).

Moreover, the innovation process $\mathbf{e}^s(t)$ is the error of projecting $\mathbf{y}(t)$ onto the linear space spanned by the products of the past values of \mathbf{y} , \mathbf{u} and the scheduling process $\boldsymbol{\mu}$, as defined in (12).

5. REALIZATION ALGORITHMS

In this section, we first recall the basis reduced Ho-Kalman realization algorithm for deterministic LPV state-space representations. In turn, this algorithm will be used for covariance realization algorithms for estimating LPV-SSA representations of \mathbf{y}^d , \mathbf{y}^s , presented in Section 5.2–5.3.

5.1 Basis reduced Ho-Kalman realization algorithm

Recall from Petreczky et al. (2017); Cox et al. (2018) that a deterministic LPV-SSA representation (with affine dependence) is a system of the form

$$\begin{aligned} x(t+1) &= \sum_{i=1}^{n_\mu} (A_i x(t) + B_i u(t)) \mu_i(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (13)$$

where A_i, B_i, C, D are matrices of suitable dimensions, $x : \mathbb{Z} \rightarrow \mathbb{R}^{n_x}$ is the state trajectory $u : \mathbb{Z} \rightarrow \mathbb{R}^{n_u}$ is the input trajectory $y : \mathbb{Z} \rightarrow \mathbb{R}^{n_y}$ is the output trajectory. In order to avoid technical problems, we assume that x, u, y all have finite support, i.e. there exist a $t_0 \in \mathbb{Z}$, such that $x(s) = 0, y(s) = 0, u(s) = 0$ for all $s < t_0$. We identify a deterministic LPV-SSA of the form (13) with the tuple $\mathcal{S} = (\{A_\sigma, B_\sigma\}_{\sigma \in \Sigma}, C, D)$. The number n_x is called the dimension of \mathcal{S} . The *sub-Markov parameters* of $\mathcal{S} = (\{A_\sigma, B_\sigma\}_{\sigma \in \Sigma}, C, D)$ are the values of the map $M_{\mathcal{S}} : \Sigma^* \rightarrow \mathbb{R}^{n_y \times n_u}$, such that for all $w \in \Sigma^*$,

$$M_{\mathcal{S}}(w) = \begin{cases} CA_s B_\sigma, & w = \sigma s, \sigma \in \Sigma, s \in \Sigma^* \\ D, & w = \epsilon \end{cases} \quad (14)$$

We will refer to $M_{\mathcal{S}}$ as the *sub-Markov function of the deterministic LPV-SSA representation of \mathcal{S}* . From Petreczky et al. (2017) it then follows that two deterministic LPV-SSA representations $\mathcal{S}_1, \mathcal{S}_2$ have the same input-output behavior, if and only if their sub-Markov parameters are equal, i.e., $M_{\mathcal{S}_1} = M_{\mathcal{S}_2}$. Moreover, the sub-Markov parameters can be determined from the input-output behavior.

Below we recall from Cox et al. (2018) an adaptation of this Ho-Kalman-like algorithm, which uses sub-Markov parameters to compute a deterministic LPV-SSA representation. In order to present the algorithm, we present the notion of *n-selection*. Let us define the set Σ^n as the set of all words $w \in \Sigma^*$ of length less than or equal to n , i.e., $\Sigma^n = \{w \in \Sigma^* \mid |w| \leq n\}$.

Definition 5. (Selection). We define (n, n_y, n_u) -selection as a pair (α, β) such that

1. $\alpha \subseteq \Sigma^n \times \{1, 2, \dots, n_y\}$ and $\beta \subseteq \Sigma \times \Sigma^n \times \{1, 2, \dots, n_u\}$
2. $\text{card}(\alpha) = \text{card}(\beta) = n$, where card denotes cardinality of the set.

We will fix the following ordering of α and β .

$$\alpha = \{(u_i, k_i)\}_{i=1}^n, \quad \beta = \{(\sigma_j, v_j, l_j)\}_{j=1}^n, \quad (15)$$

$u_i \in \Sigma^n, k_i \in \{1, 2, \dots, n_y\}, \sigma_j \in \Sigma, v_j \in \Sigma^n, l_j \in \{1, 2, \dots, n_u\}$

Example 1. Consider $n=2$, number of outputs and inputs $n_y = n_u = 2$, and scheduling signal dimension $n_\mu = 2$, we have, $\Sigma^n = \{\epsilon, 1, 2, 11, 12, 21, 22\}$. Then, one of the *n-selection* pair (α, β) can be chosen as, for e.g., $\alpha = \{(u_1, k_1), (u_2, k_2)\} = \{(\epsilon, 1), (11, 2)\}$ and $\beta = \{(\sigma_1, v_1, l_1), (\sigma_2, v_2, l_2)\} = \{(1, 21, 1), (2, 22, 2)\}$.

Let $M : \Sigma^* \rightarrow \mathbb{R}^{n_y \times n_u}$ be a map, values of which represent potential sub-Markov parameters (14) of an LPV-SSA. Let us now define the Hankel matrix $\mathcal{H}_{\alpha, \beta}^M \in \mathbb{R}^{n \times n}$ as follows:

$i, j = 1, \dots, n$, the (i, j) -th element of $\mathcal{H}_{\alpha, \beta}^M$ is of the form

$$[\mathcal{H}_{\alpha, \beta}^M]_{i, j} = [M(\sigma_j v_j u_i)]_{k_i, l_j}, \quad (16)$$

$[M(\sigma_j v_j u_i)]_{k_i, l_j}$ denotes the entry of $M(\sigma_j v_j u_i)$ on the k_i -th row and l_j -th column, and $(u_i, k_i) \in \alpha, (\sigma_j, v_j, l_j) \in \beta$ are as in the ordering of (15).

In addition, we define the σ -shifted Hankel-matrix $\mathcal{H}_{\sigma, \alpha, \beta}^M \in \mathbb{R}^{n \times n}$ as follows: its i, j -th entry is given by

$$[\mathcal{H}_{\sigma, \alpha, \beta}^M]_{i, j} = [M(\sigma_j v_j \sigma u_i)]_{k_i, l_j}. \quad (17)$$

Moreover, let us define Hankel matrices $\mathcal{H}_{\alpha, \sigma}^M \in \mathbb{R}^{n \times n_u}$ and $\mathcal{H}_{\beta}^M \in \mathbb{R}^{n_y \times n}$ as follows

$$[\mathcal{H}_{\alpha, \sigma}^M]_{i, j} = [M(\sigma u_i)]_{k_i, j}, \quad j = 1, \dots, n_u \quad (18)$$

$$[\mathcal{H}_{\beta}^M]_{i, j} = [M(\sigma_j v_j)]_{i, l_j}, \quad i = 1, \dots, n_y \quad (19)$$

Consider the model matrix computations summarized in Algorithm 1, using Hankel matrices and selections.

Algorithm 1 Deterministic realization: Matrix computations using Hankel matrices and *n-selection*

Input: (n, n_y, n_u) -selection (α, β) ; Hankel matrix $\mathcal{H}_{\alpha, \beta}^M$ (16); shifted Hankel-matrix $\mathcal{H}_{\sigma, \alpha, \beta}^M$ (17); Hankel matrices $\mathcal{H}_{\alpha, \sigma}^M$ and \mathcal{H}_{β}^M defined in (18)-(19) respectively, and $M(\epsilon)$.

1. Compute the matrices

$$\begin{aligned} \hat{A}_\sigma &= (\mathcal{H}_{\alpha, \beta}^M)^{-1} \mathcal{H}_{\sigma, \alpha, \beta}^M, \\ \hat{B}_\sigma &= (\mathcal{H}_{\alpha, \beta}^M)^{-1} \mathcal{H}_{\alpha, \sigma}^M, \quad \hat{C} = \mathcal{H}_{\beta}^M \end{aligned}$$

2. Compute $\hat{D} = M(\epsilon)$.

Output: Matrices $(\{\hat{A}_\sigma, \hat{B}_\sigma\}_{\sigma \in \Sigma}, \hat{C}, \hat{D})$

Lemma 3. (Adapted from (Cox et al., 2018)). Let the (n, n_y, n_u) -selection (α, β) be such that $\text{rank}(\mathcal{H}_{\alpha, \beta}^M) = n$, and assume that there exists a deterministic LPV-SSA representation \mathcal{S}_* of dimension n such that $M = M_{\mathcal{S}_*}$. Then the tuple $\hat{\mathcal{S}} = (\{\hat{A}_\sigma, \hat{B}_\sigma\}_{\sigma \in \Sigma}, \hat{C}, \hat{D})$, returned by Algorithm 1, when applied to the matrices $\mathcal{H}_{\alpha, \beta}^M, \mathcal{H}_{\sigma, \alpha, \beta}^M, \mathcal{H}_{\alpha, \sigma}^M, \mathcal{H}_{\beta}^M$ ((16)-(19)) and $M(\epsilon)$, is a minimal dimensional deterministic LPV-SSA representation such that $M_{\hat{\mathcal{S}}} = M$, i.e. $M(\sigma w) = \hat{C} \hat{A}_w \hat{B}_\sigma$ for all $w \in \Sigma^*$.

5.2 Correlation analysis: finding an LPV-SSA representation of \mathbf{y}^d

In this section, we describe an adaptation of the *correlation analysis* (CRA) method (Cox et al., 2015, 2018) for finding a stationary LPV-SSA representation of \mathbf{y}^d with noise process \mathbf{u} .

Let us define the map $\Psi_{\mathbf{u},\mathbf{y}} : \Sigma^* \rightarrow \mathbb{R}^{n_y \times n_u}$ as follows

$$\Psi_{\mathbf{u},\mathbf{y}}(w) = \begin{cases} \frac{1}{\sqrt{p_w}} E[\mathbf{y}(t)(\mathbf{z}_w^{\mathbf{u}}(t))^T] \Lambda_u^{-1} & \forall w \in \Sigma^+ \\ E[\mathbf{y}(t)\mathbf{u}^T(t)] \Lambda_u^{-1} & w = \epsilon \end{cases} \quad (20)$$

where we recall from Assumption 2, $\Lambda_u = \text{var}(\mathbf{u})$.

It turns out that if \mathbf{y} has a stationary LPV-SSA representation with input \mathbf{u} , then $\Psi_{\mathbf{u},\mathbf{y}}$ is the sub-Markov function of a deterministic LPV-SSA representation.

Lemma 4. Assume that $(\{A_\sigma, B_\sigma\}, C, D, \mathbf{x}, \mathbf{u})$ is a stationary LPV-SSA representation (without inputs, Definition 2) of \mathbf{y}^d . Then $\Psi_{\mathbf{u},\mathbf{y}}$ in (20) equals the sub-Markov function $M_{\mathcal{S}}$ (14) of the deterministic LPV-SSA representation $\mathcal{S} = (\{A_\sigma, B_\sigma\}_{\sigma \in \Sigma}, C, D)$. Conversely, if $\mathcal{S} = (\{A_\sigma, B_\sigma\}_{\sigma \in \Sigma}, \hat{C}, \hat{D})$ is a deterministic LPV-SSA representation such that its sub-Markov function $M_{\mathcal{S}}$ equals $\Psi_{\mathbf{u},\mathbf{y}}$ and it is minimal dimensional among such deterministic LPV-SSA representations, then $(\{\hat{A}_\sigma, \hat{B}_\sigma\}, \hat{C}, \hat{D}, \mathbf{x}, \mathbf{u})$ is a stationary LPV-SSA representation (without inputs) of \mathbf{y}^d .

The proof of Lemma 4 is presented in Mejari and Petreczky (2019b).

Hence, we can adapt the *basis reduced* Ho-Kalman realization algorithm as described in Algorithm 2. It is

Algorithm 2 Realization of \mathbf{y}^d : Computing an LPV-SSA representation of \mathbf{y}^d using covariances and n -selection.

Input: (n, n_y, n_u) -selection (α, β) of the form (15); $\Psi_{\mathbf{u},\mathbf{y}}(\sigma_j v_j u_i)_{k_i, l_j}$, $\Psi_{\mathbf{u},\mathbf{y}}(\sigma u_i)_{k_i, l}$, $\Psi_{\mathbf{u},\mathbf{y}}(\sigma_j v_j)_{r, j}$, $i, j = 1, \dots, n_x$, $l = 1, \dots, n_u$, $r = 1, \dots, n_y$, $\Psi_{\mathbf{u},\mathbf{y}}(\epsilon)$.

1. Construct the matrices $\mathcal{H}_{\alpha, \beta}^{\Psi_{\mathbf{u},\mathbf{y}}}$, $\mathcal{H}_{\sigma, \alpha, \beta}^{\Psi_{\mathbf{u},\mathbf{y}}}$, $\mathcal{H}_{\alpha, \sigma}^{\Psi_{\mathbf{u},\mathbf{y}}}$ and $\mathcal{H}_{\beta}^{\Psi_{\mathbf{u},\mathbf{y}}}$, by replacing M with $\Psi_{\mathbf{u},\mathbf{y}}$ in (16)–(19).
2. Apply Algorithm 1 to $\mathcal{H}_{\alpha, \beta}^{\Psi_{\mathbf{u},\mathbf{y}}}$, $\mathcal{H}_{\sigma, \alpha, \beta}^{\Psi_{\mathbf{u},\mathbf{y}}}$, $\mathcal{H}_{\alpha, \sigma}^{\Psi_{\mathbf{u},\mathbf{y}}}$, $\mathcal{H}_{\beta}^{\Psi_{\mathbf{u},\mathbf{y}}}$, $M(\epsilon) = \Psi_{\mathbf{u},\mathbf{y}}(\epsilon)$. Denote by $(\{\hat{A}_\sigma, \hat{B}_\sigma\}_{\sigma \in \Sigma}, \hat{C}, \hat{D})$ the matrices returned by Algorithm 1.

Output: Matrices $(\{\hat{A}_\sigma, \hat{B}_\sigma\}_{\sigma \in \Sigma}, \hat{C}, \hat{D})$

clear from Lemma 4 and Lemma 3 that Algorithm 2 is correct.

5.3 Covariance realization algorithm

In this section, we adapt the realization algorithm from Mejari and Petreczky (2019a) to estimate the stochastic part (7) of a LPV-SSA representation.

Define the *covariance sequence* $\Psi_{\mathbf{y}^s} : \Sigma^* \rightarrow \mathbb{R}^{n_y \times n_y}$, where $\Psi_{\mathbf{y}^s}(\epsilon) = I_{n_y}$, and for all $w \in \Sigma^+$,

$$\Psi_{\mathbf{y}^s}(w) = E[\mathbf{y}^s(t)(\mathbf{z}_w^{\mathbf{y}^s}(t))^T] \quad (21)$$

If \mathbf{y}^s has a stationary LPV-SSA representation, then $\Psi_{\mathbf{y}^s}$ is a sub-Markov function of a suitable deterministic LPV-SSA representation, Petreczky and Vidal (2018); Mejari and Petreczky (2019a). Conversely, from a deterministic LPV-SSA representation, sub-Markov function of which equals $\Psi_{\mathbf{y}^s}$ a stationary LPV-SSA representation can be computed.

Lemma 5. If $\mathcal{S} = (\{\hat{A}_\sigma, \hat{G}_\sigma\}_{\sigma \in \Sigma}, \hat{C}, I_{n_y})$ is a minimal dimensional deterministic LPV-SSA representation such

that $M_{\mathcal{S}} = \Psi_{\mathbf{y}^s}$, then $(\{\hat{A}_\sigma, \hat{K}_\sigma\}_{\sigma \in \Sigma}, \hat{C}, I_{n_y}, \hat{\mathbf{x}}, \mathbf{e}^s)$ is a stationary LPV-SSA representation of \mathbf{y}^s in forward innovation form, where $\hat{A}_\sigma^s = \frac{1}{\sqrt{p_\sigma}} \hat{A}_\sigma$, $\hat{C}^s = \hat{C}$, $\hat{K}_\sigma = \lim_{i \rightarrow \infty} \hat{K}_\sigma^i$, and $\{\hat{K}_\sigma^i\}_{\sigma \in \Sigma, i \in \mathbb{N}}$ satisfies the following recursion

$$\begin{aligned} \hat{P}_\sigma^{i+1} &= \sum_{\sigma_1 \in \Sigma} p_{\sigma_1} \left(\hat{A}_{\sigma_1}^s \hat{P}_\sigma^i (\hat{A}_{\sigma_1}^s)^T + \hat{K}_{\sigma_1} \hat{Q}_{\sigma_1}^i \hat{K}_{\sigma_1}^T \right) \\ \hat{Q}_\sigma^i &= p_\sigma E[\mathbf{z}_\sigma^{\mathbf{y}^s}(t)(\mathbf{z}_\sigma^{\mathbf{y}^s}(t))^T] - \hat{C}^s \hat{P}_\sigma^i (\hat{C}^s)^T \\ \hat{K}_\sigma^i &= \left(\hat{G}_\sigma \sqrt{p_\sigma} - \hat{A}_\sigma^s \hat{P}_\sigma^i (\hat{C}^s)^T \right) \left(\hat{Q}_\sigma^i \right)^{-1} \end{aligned} \quad (22)$$

with $\hat{P}_\sigma^0 = 0$. Moreover, $E[\mathbf{e}^s(t)(\mathbf{e}^s(t))^T \mu_\sigma^2(t)] = \hat{Q}_\sigma = \lim_{i \rightarrow \infty} \hat{Q}_\sigma^i$, $E[\hat{\mathbf{x}}(t)\hat{\mathbf{x}}^T(t) \mu_\sigma^2] = \hat{P}_\sigma = \lim_{i \rightarrow \infty} \hat{P}_\sigma^i$ for all $\sigma \in \Sigma$.

The proof of Lemma 5 can be found in Petreczky and Vidal (2018); Mejari and Petreczky (2019a). From Lemma 5, it follows that we can use the basis reduced Kalman-Ho realization algorithm Algorithm 2, as described in Algorithm 3, in order to compute LPV-SSA representation of \mathbf{y}^s .

Algorithm 3 Realization of \mathbf{y}^s : Computing an LPV-SSA representation of \mathbf{y}^s using covariances and n -selection.

Input: (n, n_y, n_y) -selection $(\bar{\alpha}, \bar{\beta})$ of the form of the form (15); $\Psi_{\mathbf{y}^s}(\sigma_j v_j u_i)_{k_i, l_j}$; $\Psi_{\mathbf{y}^s}(\sigma u_i)_{k_i, l}$; $\Psi_{\mathbf{y}^s}(\sigma_j v_j)_{r, j}$, $i, j = 1, \dots, n$; $l, r = 1, \dots, n_y$, $\{E[\mathbf{z}_\sigma^{\mathbf{y}^s}(t)(\mathbf{z}_\sigma^{\mathbf{y}^s}(t))^T]\}_{\sigma \in \Sigma}$; number of maximal iterations $\mathcal{I} > 0$.

1. Construct the matrices $\mathcal{H}_{\bar{\alpha}, \bar{\beta}}^{\Psi_{\mathbf{y}^s}}$, $\mathcal{H}_{\sigma, \bar{\alpha}, \bar{\beta}}^{\Psi_{\mathbf{y}^s}}$, $\mathcal{H}_{\bar{\alpha}, \sigma}^{\Psi_{\mathbf{y}^s}}$, $\mathcal{H}_{\bar{\beta}}^{\Psi_{\mathbf{y}^s}}$ by replacing M with $\Psi_{\mathbf{y}^s}$ in (16)–(19).
2. Apply Algorithm 1 to $\mathcal{H}_{\bar{\alpha}, \bar{\beta}}^{\Psi_{\mathbf{y}^s}}$, $\mathcal{H}_{\sigma, \bar{\alpha}, \bar{\beta}}^{\Psi_{\mathbf{y}^s}}$, $\mathcal{H}_{\bar{\alpha}, \sigma}^{\Psi_{\mathbf{y}^s}}$, $\mathcal{H}_{\bar{\beta}}^{\Psi_{\mathbf{y}^s}}$, $M(\epsilon) = I_{n_y}$. Denote by $(\{\hat{A}_\sigma, \hat{G}_\sigma\}_{\sigma \in \Sigma}, \hat{C}, I_{n_y})$ the matrices returned by Algorithm 1.
3. Define $\hat{A}_\sigma^s = \frac{1}{\sqrt{p_\sigma}} \hat{A}_\sigma$, $\sigma \in \Sigma$, $\hat{C}^s = \hat{C}$.
4. Compute $\{\hat{K}_\sigma^i, \hat{Q}_\sigma^i, \hat{P}_\sigma^i\}_{i=1}^{\mathcal{I}}$ using the recursion (22).

Output: Matrices $(\{\hat{A}_\sigma^s, \hat{G}_\sigma, \hat{K}_\sigma^{\mathcal{I}}, \hat{Q}_\sigma^{\mathcal{I}}, \hat{P}_\sigma^{\mathcal{I}}\}_{\sigma \in \Sigma}, \hat{C}^s)$

It is clear from Lemma 5 and Lemma 3 that Algorithm 3 is correct.

6. IDENTIFICATION ALGORITHM

In this section, we formulate an identification algorithm based on stochastic realization Algorithms 2–3 and selections, for N -length observation sequence of outputs, inputs and scheduling signals, as detailed in Algorithm 4. Intuitively, the main idea behind Algorithm 4 is to estimate the covariances $\Psi_{\mathbf{u},\mathbf{y}}$, $\Psi_{\mathbf{y}^s}$ and $E[\mathbf{z}_\sigma^{\mathbf{y}}(t)(\mathbf{z}_\sigma^{\mathbf{y}}(t))^T]$ from the observed data and then apply Algorithms 2–3 to the thus estimated covariances. More specifically, the following assumptions are made:

Assumption 3. (1) The n_x -selection pair (α, β) and $(\bar{\alpha}, \bar{\beta})$ are such that $\text{rank } \mathcal{H}_{\alpha, \beta}^{\Psi_{\mathbf{u},\mathbf{y}}} = n_x$, $\text{rank } \mathcal{H}_{\bar{\alpha}, \bar{\beta}}^{\Psi_{\mathbf{y}^s}} = n_x$, where n_x is the state-space dimension of a minimal LPV-SSA realization of \mathbf{y} .

(2) The process $(\mathbf{y}, \mathbf{u}, \{\mu_w\}_{w \in \Sigma^+})$ is ergodic and there exist sample paths $\mathbf{y} : \mathbb{Z} \rightarrow \mathbb{R}^{n_y}$, $\mathbf{u} : \mathbb{Z} \rightarrow \mathbb{R}^{n_u}$ and $\mu : \mathbb{Z} \rightarrow \mathbb{R}^{n_\mu}$ of the processes \mathbf{y} , \mathbf{u} and μ respectively

such that $\{y(t), u(t), \{\mu_\sigma(t)\}_{\sigma \in \Sigma}\}_{t=1}^N$ is observed and the following holds: for all $w \in \Sigma^*$, $\sigma \in \Sigma$,

$$\Psi_{\mathbf{u}, \mathbf{y}}^N(w) = \frac{1}{\sqrt{p_w}} \left(\frac{1}{N} \sum_{t=|w|}^N y(t) (z_w^u(t))^T \right) \Lambda_u^{-1}, \quad w \in \Sigma^*$$

$$\Lambda_{\sigma w}^{\mathbf{y}, N} = \frac{1}{N} \sum_{t=|w|}^N y(t) (z_{\sigma w}^y(t))^T, \quad T_{\sigma, \sigma}^{\mathbf{y}, N} = \frac{1}{N} \sum_{t=1}^N z_\sigma^y(t) (z_\sigma^y(t))^T$$

Then for all $w \in \Sigma^*$, $\sigma \in \Sigma$,

$$\Psi_{\mathbf{u}, \mathbf{y}}(w) = \lim_{N \rightarrow \infty} \Psi_{\mathbf{u}, \mathbf{y}}^N(w), \quad E[\mathbf{z}_\sigma^y(t) (\mathbf{z}_\sigma^y(t))^T] = \lim_{N \rightarrow \infty} T_{\sigma, \sigma}^{\mathbf{y}, N}$$

$$E[\mathbf{y}(t) (\mathbf{z}_{\sigma w}^y(t))^T] = \lim_{N \rightarrow \infty} \Lambda_{\sigma w}^{\mathbf{y}, N},$$

where, for all $w = \sigma_1 \sigma_2 \cdots \sigma_r \in \Sigma^+$, $r > 0$, we have,

$$\mu_w(t) = \mu_{\sigma_1}(t - k + 1) \mu_{\sigma_2}(t - k + 2) \cdots \mu_{\sigma_r}(t)$$

$$z_w^u(t) = u(t - |w|) \mu_w(t - 1) \frac{1}{\sqrt{p_w}}, \quad z_\epsilon^u(t) = u(t)$$

$$z_w^y(t) = y(t - |w|) \mu_w(t - 1) \frac{1}{\sqrt{p_w}}$$

Lemma 6. (Consistency). With the Assumption 3 the result of Algorithm 4 satisfies the following:

$$\tilde{K}_\sigma = \lim_{\mathcal{I} \rightarrow \infty} \lim_{N \rightarrow \infty} \tilde{K}_\sigma^{N, \mathcal{I}}, \quad \tilde{A}_\sigma = \lim_{N \rightarrow \infty} \tilde{A}_\sigma^N,$$

$$\tilde{B}_\sigma = \lim_{N \rightarrow \infty} \tilde{B}_\sigma^N, \quad \tilde{C} = \lim_{N \rightarrow \infty} \tilde{C}^N, \quad \tilde{D} = \lim_{N \rightarrow \infty} \tilde{D}^N$$

and $(\{\tilde{A}_\sigma, \tilde{B}_\sigma, \tilde{K}_\sigma\}_{\sigma=1}^{n_\mu}, \tilde{C}, \tilde{D}, \hat{\mathbf{x}}, \mathbf{e}^s)$ is a stationary LPV-SSA representation of $(\mathbf{y}, \mathbf{u}, \boldsymbol{\mu})$, and $E[\mathbf{e}^s(t) (\mathbf{e}^s(t))^T \boldsymbol{\mu}_\sigma^2(t)] = \lim_{\mathcal{I} \rightarrow \infty} \lim_{N \rightarrow \infty} \tilde{Q}_\sigma^{N, \mathcal{I}}$, $\sigma \in \Sigma$.

The proof of Lemma 6 is presented in Mehari and Petreczky (2019b).

Remark 1. (Alternative way of computing $\Psi_{\mathbf{y}^s}^N$). An alternative way of estimating the covariances $\Psi_{\mathbf{y}^s}$ and $E[\mathbf{z}_\sigma^y(t) (\mathbf{z}_\sigma^y(t))^T]_{\sigma \in \Sigma}$ is to use the matrices $\mathcal{S} = (\{\tilde{A}_\sigma^d, \tilde{B}_\sigma^d\}_{\sigma \in \Sigma}, \tilde{C}^d, \tilde{D}^d)$ to approximate the sample paths y^d , y^s of \mathbf{y}^d and \mathbf{y}^s by $\hat{y}^d(t) = \tilde{D}^d u(t) + \sum_{v \in \Sigma^*, \sigma \in \Sigma, |v| < t-1} \tilde{C}^d \tilde{A}_v^d \tilde{B}_\sigma^d z_{\sigma v}^u(t)$, and $\hat{y}^s(t) = y(t) - \hat{y}^d(t)$ and define

$$\Psi_{\mathbf{y}^s}^N(w) = \frac{1}{N} \sum_{t=|w|}^N \hat{y}^s(t) z_w^{\hat{y}^s}(t), \quad w \in \Sigma^+$$

$$T_{\sigma, \sigma}^N = \frac{1}{N} \sum_{t=|w|}^N z_\sigma^{\hat{y}^s}(t) (z_\sigma^{\hat{y}^s}(t))^T, \quad \sigma \in \Sigma \quad (25)$$

where $z_w^{\hat{y}^s}(t) = \hat{y}^s(t - |w|) \mu_w(t - 1) \frac{1}{\sqrt{p_w}}$ for all $w \in \Sigma^+$. We can then view $\Psi_{\mathbf{y}^s}^N(w)$ as an approximation of $\Psi_{\mathbf{y}^s}(w)$, and $T_{\sigma, \sigma}^N$ is an approximation of $E[\mathbf{z}_\sigma^y(t) (\mathbf{z}_\sigma^y(t))^T]_{\sigma \in \Sigma}$. We could modify Algorithm 4 by replacing (23) with (25). We conjecture that Lemma 6 will remain true for the modified algorithm.

7. NUMERICAL EXAMPLE

In this section, we present a numerical example to test the effectiveness of our algorithm. All computations are carried out on an i5 1.8-GHz Intel core processor with 8 GB of RAM running MATLAB R2018a.

Algorithm 4 Identification of stochastic LPV-SSA from observed data.

Input: Observations sequence $\{y(t), u(t), \{\mu_\sigma(t)\}_{\sigma \in \Sigma}\}_{t=1}^N$, and n_x -selection (α, β) and $(\bar{\alpha}, \bar{\beta})$; $\{p_\sigma\}_{\sigma \in \Sigma}$, Λ_u , maximum number of iterations $\mathcal{I} > 0$.

1. Compute empirical covariances $\Psi_{\mathbf{u}, \mathbf{y}}^N(w)$, for every $w \in \Sigma^+$, such that $w = ivu$ or $w = iv\sigma u$ or $w = iv$ or $w = \sigma u$ for some words $v, u \in \Sigma^*$, $\sigma \in \Sigma$, $(u, k) \in \alpha$, $(i, v, l) \in \beta$ for some $k = 1, \dots, n_y$, $l = 1, \dots, n_u$, $\forall w \in \Sigma^+$.
2. Run Algorithm 2 with empirical covariances $\Psi_{\mathbf{u}, \mathbf{y}}^N(w)$, instead of the covariances $\Psi_{\mathbf{u}, \mathbf{y}}$. Denote the result returned by Algorithm 2 by $\mathcal{S} = (\{\tilde{A}_\sigma^d, \tilde{B}_\sigma^d\}_{\sigma \in \Sigma}, \tilde{C}^d, \tilde{D}^d)$.
3. Compute approximate covariances:
$$\Psi_{\mathbf{y}^s}^N(\sigma w) = \Lambda_{\sigma w}^{\mathbf{y}, N} - \Lambda_{\mathcal{S}}(\sigma w)$$

$$\Lambda_{\mathcal{S}}(\sigma w) = \frac{1}{\sqrt{p_{\sigma w}}} \tilde{C}^d \tilde{A}_w^d (\tilde{A}_\sigma^d \tilde{P}_\sigma (\tilde{C}^d)^T + \tilde{B}_\sigma^d \Lambda_u)$$

$$T_{\sigma, \sigma}^N = T_{\sigma, \sigma}^{\mathbf{y}, N} - T_{\sigma, \sigma, \mathcal{S}}$$

$$T_{\sigma, \sigma, \mathcal{S}} = \frac{1}{p_\sigma} (\tilde{C}^d \tilde{P}_\sigma (\tilde{C}^d)^T + \Lambda_u)$$

for all $\sigma \in \Sigma$ and for every $w \in \Sigma^+$, such that $w = ivu$ or $w = iv$ or $w = iu$ or $w = iv\sigma u$ for some words $v, u \in \Sigma^*$, $i, \sigma \in \Sigma$, $(u, k) \in \bar{\alpha}$, $(v, l) \in \bar{\beta}$ for some $k, l = 1, \dots, n_y$, for all $w \in \Sigma^+$. Here, \tilde{P}_σ is the unique solution to the following Sylvester equation

$$\tilde{P}_\sigma = p_\sigma \sum_{\sigma_1 \in \Sigma} \left(\tilde{A}_{\sigma_1}^d \tilde{P}_{\sigma_1} (\tilde{A}_{\sigma_1}^d)^T + \tilde{B}_{\sigma_1}^d \Lambda_u (\tilde{B}_{\sigma_1}^d)^T \right). \quad (24)$$

4. Run Algorithm 3, with the empirical covariances $\Psi_{\mathbf{y}^s}^N$ instead of $\Psi_{\mathbf{y}^s}$, and $\{T_{\sigma, \sigma}^N\}_{\sigma \in \Sigma}$ instead of $\{E[\mathbf{z}_\sigma^y(t) (\mathbf{z}_\sigma^y(t))^T]_{\sigma \in \Sigma}\}$. Denote the result returned by Algorithm 3 by $(\{\tilde{A}_\sigma^s, \tilde{K}_\sigma^{L, s}, \tilde{Q}_\sigma^L, \tilde{P}_\sigma^L\}_{\sigma \in \Sigma}, \tilde{C}^s)$.
5. The estimated model matrices of LPV-SSA (1) are given by $\tilde{A}_\sigma^N = \text{diag}(\tilde{A}_\sigma^d, \tilde{A}_\sigma^s)$, $\tilde{B}_\sigma^N = [(\tilde{B}_\sigma^d)^T \ 0^T]^T$, $\tilde{K}_\sigma^{N, \mathcal{I}} = [0^T \ (\tilde{K}_\sigma^{L, s})^T]^T$, $\forall \sigma \in \Sigma$, $\tilde{C}^N = [\tilde{C}^d \ \tilde{C}^s]$, $\tilde{D}^N = \tilde{D}^d$, $\tilde{Q}_\sigma^{N, \mathcal{I}} = \tilde{Q}_\sigma^L$.

Output: Estimates $\{\tilde{A}_\sigma^N, \tilde{B}_\sigma^N, \tilde{K}_\sigma^{N, \mathcal{I}}, \tilde{Q}_\sigma^{N, \mathcal{I}}\}_{\sigma \in \Sigma}, \tilde{C}^N, \tilde{D}^N$

The quality of the match between estimated and true outputs is quantified on a noise-free validation data of length N_{val} via *Best Fit Rate* (BFR) and *Variance Accounted For* (VAF) criterion defined for each output channel y_i , $i = 1, \dots, n_y$, as $\text{BFR}_{y_i} = \max \left\{ 1 - \sqrt{\frac{\sum_{t=1}^{N_{\text{val}}} (y_i(t) - \hat{y}_i(t))^2}{\sum_{t=1}^{N_{\text{val}}} (y_i(t) - \bar{y}_i)^2}}, 0 \right\} \times 100\%$, $\text{VAF}_{y_i} = \max \left\{ 1 - \frac{\text{var}(y_i - \hat{y}_i)}{\text{var}(y_i)}, 0 \right\} \times 100\%$, where \hat{y}_i denotes the simulated one-step ahead model output and \bar{y}_i denotes the sample mean of the output over the validation set.

The LPV-SSA representation in form (1) is used for data generation with following matrices:

$$A_1 = \begin{bmatrix} 0.4 & 0.4 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.4 & 0.4 \\ 0 & 0.4 & 0.4 \end{bmatrix},$$

$$B_1 = [1 \ 1 \ 1]^T, \quad B_2 = [1 \ 0 \ 1]^T,$$

Table 1. BFR and VAF on a noise-free validation data Algorithm 4

BFR	93.56 %
VAF	99.58 %

Table 2. True vs estimated sub-Markov parameters

Markov parameters	True value	Estimated value
CA_1B_1	0.80	0.7957
CA_1B_2	0.40	0.3914
$CA_1^2B_1$	0.32	0.3147
$CA_1A_2B_2$	0.16	0.1549
$CA_1^3B_1$	0.12	0.1093

$K_1 = [-0.036 \ 0 \ 1]^T$, $K_2 = [0 \ 0.015 \ 1.17]^T$, $C = [1 \ 0 \ 0]$, which corresponds to state-dimension $n_x = 3$, output dimension $n_y = 1$, and scheduling signal dimension $n_\mu = 2$ with $\Sigma = \{1, 2\}$.

Training and noise free validation output sequences of length $N = 100000$ and $N_{\text{val}} = 100000$, respectively, are generated using a white-noise input process \mathbf{u} with uniform distribution $\mathcal{U}(-1.5, 1.5)$ and an independent scheduling signal process $\boldsymbol{\mu} = [\boldsymbol{\mu}_1 \ \boldsymbol{\mu}_2]$ such that $\boldsymbol{\mu}_1(t) = 1$ and $\boldsymbol{\mu}_2(t)$ is a white-noise process with uniform distribution $\mathcal{U}(-1.5, 1.5)$. This corresponds to the parameter values $\{p_\sigma\}_{\sigma \in \{1,2\}}$ to be $p_1 = E[\boldsymbol{\mu}_1^2(t)] = 1$ and $p_2 = E[\boldsymbol{\mu}_2^2(t)] = 0.75$. The standard deviation of the white Gaussian noise \mathbf{e} corrupting the training output is 1, i.e., $\mathbf{e} \sim \mathcal{N}(0, 1)$. This corresponds to the *Signal-to-Noise Ratio* SNR = $10 \log \frac{\sum_{t=1}^N (y(t) - e(t))^2}{\sum_{t=1}^N e^2(t)} = 4.7$ dB.

We run the version of Algorithm 4 explained in Remark 1, with $\mathcal{I} = 50$ iterations and with the following n -selection pairs (α, β) and $(\bar{\alpha}, \bar{\beta})$, with $n = 3$,

$\alpha = \{(\epsilon, 1), (1, 1), (21, 1)\}$, $\beta = \{(2, \epsilon, 1), (1, 2, 1), (2, 21, 1)\}$,
 $\bar{\alpha} = \{(\epsilon, 1), (1, 1), (21, 1)\}$, $\bar{\beta} = \{(1, \epsilon, 1), (1, 2, 1), (1, 21, 1)\}$,

which are used to choose corresponding entries of the Hankel matrices. The mean time taken to run the algorithm is 1.55 sec. The validation result using one-step ahead predicted outputs \hat{y} are reported in Table 1, and true vs estimated sub-Markov parameters are reported in Table 2. The results show a good match between estimated model output w.r.t. true system output.

8. CONCLUSION

In this paper, we formulated a realization algorithm and an efficient identification algorithm for stochastic LPV-SSA representations with inputs, by combining correlation analysis method with a stochastic realization based identification algorithm. The proposed algorithm provides a computationally efficient alternative to the parametric subspace approaches avoiding the curse of dimensionality.

REFERENCES

Bamieh, B.A. and Giarré, L. (2002). Identification of linear parameter-varying models. *International Journal of Robust Nonlinear Control*, 12(9), 841–853.
 Cox, P., Petreczky, M., and Tóth, R. (2018). Towards efficient maximum likelihood estimation of LPV-SS models. *Automatica*, 97(9), 392–403.

Cox, P., Tóth, R., and Petreczky, M. (2015). Estimation of LPV-SS models with static dependency using correlation analysis. In *Proc. 1st IFAC Workshop on Linear Parameter Varying Systems*, 91–96. Grenoble, France.
 dos Santos, P., Ramos, J., and de Carvalho, J. (2009). Identification of bilinear systems with white noise inputs: An iterative deterministic-stochastic subspace approach. *IEEE Transactions on Control Systems Technology*, 17(5), 1145–1153.
 Favoreel, W., De Moor, B., and Van Overschee, P. (1999). Subspace identification of bilinear systems subject to white inputs. *IEEE Transactions on Automatic Control*, 44(6), 1157–1165. doi:10.1109/9.769370.
 Felici, F., Van Wingerden, J.W., and Verhaegen, M. (2007). Subspace identification of MIMO LPV systems using a periodic scheduling sequence. *Automatica*, 43, 1684–1697.
 Laurain, V., Gilson, M., Tóth, R., and Garnier, H. (2010). Refined instrumental variable methods for identification of LPV Box–Jenkins models. *Automatica*, 46(6), 959–967.
 Mejari, M. and Petreczky, M. (2019a). Consistent and computationally efficient estimation for stochastic LPV state-space models: realization based approach. In *IEEE Conference on Decision and Control*. Nice, France.
 Mejari, M. and Petreczky, M. (2019b). Realization and identification algorithm for stochastic LPV state-space models with exogenous inputs. Technical report, arXiv:1905.10113.
 Mejari, M., Piga, D., and Bemporad, A. (2018). A bias-correction method for closed-loop identification of Linear Parameter-Varying systems. *Automatica*, 87, 128–141.
 Petreczky, M., Tóth, R., and Mercére, G. (2017). Realization theory for LPV state-space representations with affine dependence. *IEEE Transactions on Automatic Control*, 62(9), 4667–4674.
 Petreczky, M. and Vidal, R. (2018). Realization theory for a class of stochastic bilinear systems. *IEEE Transactions on Automatic Control*, 63(1), 69–84.
 Piga, D., Cox, P., Tóth, R., and Laurain, V. (2015). LPV system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53, 329–338.
 Tanelli, M., Ardagna, D., and Lovera, M. (2011). Identification of LPV state space models for autonomic web service systems. *IEEE Transactions on Control Systems Technology*, 19(1), 93–103.
 Tóth, R. (2010). *Modeling and identification of linear parameter-varying systems*. Lecture Notes in Control and Information Sciences, Vol. 403, Springer, Heidelberg.
 Tóth, R., Abbas, H., and Werner, H. (2012). On the state-space realization of LPV input-output models: Practical approaches. *IEEE Transactions on Control Systems Technology*, 20(1).
 van Wingerden, J.W. and Verhaegen, M. (2009). Subspace identification of bilinear and LPV systems for open-and closed-loop data. *Automatica*, 45(2), 372–381.
 Verdult, V. and Verhaegen, M. (2005). Kernel methods for subspace identification of multivariable LPV and bilinear systems. *Automatica*, 41, 1557–1565.