2019 18th European Control Conference (ECC)
Napoli, Italy, June 25-28, 2019

Kernelized Identification of Linear Parameter-Varying Models with Linear Fractional Representation

Manas Mejari, Dario Piga, Roland Tóth and Alberto Bemporad

Abstract— **The article presents a method for the identification of *Linear Parameter-Varying* (LPV) models in a *Linear Fractional Representation* (LFR), which corresponds to a *Linear Time-Invariant* (LTI) model connected to a scheduling variable dependency via a feedback path. A two-stage identification approach is proposed. In the first stage, *Kernelized Canonical Correlation Analysis* (KCCA) is formulated to estimate the state sequence of the underlying LPV model. In the second stage, a non-linear least squares cost function is minimized by employing a coordinate descent algorithm to estimate latent variables characterizing the LFR and the unknown model matrices of the LTI block by using the state estimates obtained at the first stage. Here, it is assumed that the structure of the scheduling variable dependent block in the feedback path is fixed. For a special case of affine dependence of the model on the feedback block, it is shown that the optimization problem in the second stage reduces to ordinary least-squares followed by a singular value decomposition.**

I. INTRODUCTION

The *Linear Parameter-Varying* (LPV) paradigm represents a natural extension of *Linear Time-Invariant (LTI)* models. The property of linearity is preserved in the dynamic relation between input and output signals, but this relation can change over time according to a measurable time-varying signal, the so called *scheduling variable*. By virtue of this scheduling signal, LPV models can describe the behavior of many time-varying and non-linear systems accurately. In the existing literature, many methods have been proposed for the identification of LPV models, both in state-space (SS) [14], [17], [19], [20] and input-output (IO) representations [2], [3], [7], [8], [10], [16]. A detailed summary of the available LPV identification approaches can be found in [15].

The main drawback of the IO approaches is that the obtained models are not well suited for controller synthesis. The controller design approaches for LPV models (e.g., [12], [21]) often require the LPV models to be in either SS representation with an affine dependency on the scheduling signal or in a *Linear Fractional Representation* (LFR) depicted in Fig. 1. While restricting SS representations to affine dependency is a necessity in polytopic control synthesis methods, it often requires masking nonlinear dependencies on measured signals as independent scheduling variables leading to significant increase of the scheduling space and conservatism of the synthesis. Compared to such representations, LFRs can easily handle rational dependencies in a simple fashion, thus providing control synthesis methods with reduced conservatism [12]. Hence, identification of LPV-LFR model structures is of significant practical importance.

Very few works have addressed the problem of identifying LPV models in an LFR form. To mention a few, in [9], it is shown that under the assumption of full state measurement, the problem of SISO LPV-LFR model identification can be solved by recursive least-squares. A prediction error method is proposed in [5] and [6], where the mean-squared prediction error is minimized using non-linear programming. In [4], an identification approach is presented for SISO LPV-LFR models with scalar scheduling variable. By suitably manipulating the scheduling signal trajectory, it is shown that, for the noise-free case, the model can be identified using convex optimization. For noisy measurements, tractable convex relaxations are proposed under the assumption that a bound on the measurement noise is known.

In this work, we present a two-stage method for the identification of LPV models with LFRs which can be seen as a step towards direct identification of LPV-LFR models without structural assumptions. The first stage consists of estimating the state sequence of the underlying LPV models using Kernel Canonical Correlation Analysis (KCCA), which has been recently introduced for LPV-SS models in [11]. In a KCCA approach, the correlation between past and future data samples is maximized in order to estimate the state sequence up to a similarity transformation. In the second stage, a cost function is minimized with a coordinate descent algorithm to estimate the latent variables z characterizing LFRs and the unknown model matrices in the forward LTI part, using the state estimates obtained at the first stage. It is shown that under the assumption of affine parametric dependency and diagonal structure of the feedback block, the unknown LTI model matrices can be computed using ordinary least square followed by a singular value decomposition. In overall, the main contributions of the paper are: (i) parametric identification of LPV state-space models is obtained with *rational* scheduling dependencies; (ii) the proposed method provides a computationally efficient alternative to the parametric state-space approaches dealing with non-affine scheduling variable dependencies which are prone to computational and dimensionality problems.

*This work was partially supported by the H2020-723248 project *DAEDALUS - Distributed control and simulation platform to support an ecosystem of digital automation developers*.

M. Mejari is with the Centre de Recherche en Informatique, Signal et Automatique de Lille, University of Lille 1, Villeneuve-d'Ascq 59651, France. manas-dilip.mejari@univ-lille.fr

D. Piga is with IDSIA Dalle Molle Institute for Artificial Intelligence SUPSI-USI, 6928 Manno, Switzerland. dario.piga@supsi.ch

R. Tóth is with the Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. r.toth@tue.nl

A. Bemporad is with IMT School for Advanced Studies Lucca, 55100 Lucca, Italy. alberto.bemporad@imtlucca.it

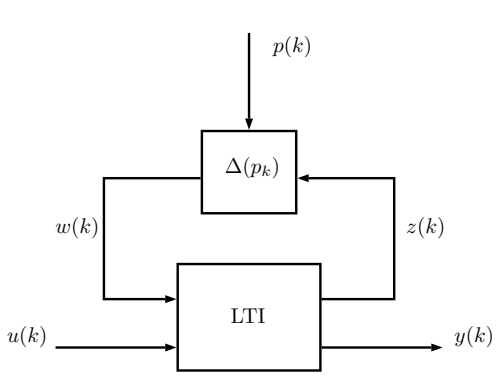

Fig. 1: Linear fractional representation of LPV systems

II. NOTATION

Let $\mathbb{R}^n$ be the set of real vectors of dimension n. The i-th element of a vector $x \in \mathbb{R}^n$ is denoted by $[x]_i$ and $\|x\|_Q^2 = x^\top Q x$ denotes the squared weighted 2-norm of x. For matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, the Kronecker product between A and B is denoted by $A \otimes B \in \mathbb{R}^{mp\times nq}$. Let $\mathbb{I}_a^b$ be the sequence of successive integers $\{a, a+1, \cdots, b\}$, with $a < b$. The Moore-Penrose pseudo-inverse of a non-square matrix A is denoted by $A^\dagger$. The notation $(A^d \diamond p)(k)$ is used to express the dynamic dependence of the matrix A on p at time k, e.g., A at time k depends on the future d samples of the signal p, i.e., $p(k), \ldots, p(k+d-1)$. Similarly, dependance on past samples is denoted in the same way.

III. PROBLEM FORMULATION

By referring to Fig. 1, we consider the following discrete time LPV data generating system in a linear fractional representation with the LTI part given by,

$$\left[\begin{array}{c} x(k+1) \\ \hline z(k) \\ y_{\mathrm{o}}(k) \end{array}\right] = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \left[\begin{array}{c} x(k) \\ \hline w(k) \\ u(k) \end{array}\right], \quad \text{(1a)}$$

where $x(k) \in \mathbb{R}^{n_{\mathrm{x}}}$ is the state, $u(k) \in \mathbb{R}^{n_{\mathrm{u}}}$ and $y_{\mathrm{o}}(k) \in \mathbb{R}^{n_{\mathrm{y}}}$ are the measured input and output of the system at time k respectively. The ouput measurements are corrupeted by zero mean additive white noise, i.e., $y(k) = y_{\mathrm{o}}(k) + e(k)$ and $\{A, \ldots, D_{22}\}$ are unknown constant martices of appropriate dimensions. The feedback path is represented by

$$w(k) = \Delta(p(k))z(k), \quad \text{(1b)}$$

where, $\Delta : \mathbb{P} \to \mathbb{R}^{n_p\times n_p}$ is a function of the scheduling variable p. The variables x, z and w are latent (auxillary) variables whose measurements are not available.

The following assumptions are made on the system (1):

A1. the structure of the feedback block Δ is known.

A2. $I - D_{11}\Delta(p(k)) \neq 0$ for all trajectories of the scheduling signal $p(k) \in \mathbb{P}$.

We now formally state the identification problem addressed in this contribution.

Problem 1: Given an N-length training dataset $\mathcal{D} = \{u(k), y(k), p(k)\}_{k=1}^N$, identify the LPV system represented by the LFR (1), by estimating the unknown matrices $\{A, \ldots, D_{22}\}$, to match the input-output behavior of the underlying data generating system.

IV. IDENTIFICATION ALGORITHM

In this section, we describe the proposed method for identification of the LPV-LFR model (1). The method consists of two stages. In the first stage, the estimate $\hat{x}(k)$ of the state sequence characterizing the LTI block is obtained using Kernel Canonical Correlation Analysis (KCCA) [11], [18]. In the second stage, the latent variable sequence $z(k)$ and the unknown matrices $\{A, \ldots, D_{22}\}$ are estimated by minimizing a non-linear least squares cost function using coordinate descent algorithm.

A. KCCA for state estimation of LPV models

In this section, we review Kernel Canonical Correlation Analysis (KCCA) [11], [18] to obtain a state sequence $\{\hat{x}(k)\}_{k=1}^N$ which is compatible with the given dataset $\mathcal{D}$.

The model (1) can be represented in a LPV state-space form. By re-writing the latent variable $z(k)$ in (1) as

$$z(k) = (I - D_{11}\Delta(p(k))^{-1}\left(C_1 x(k) + D_{12}u(k)\right), \quad (2)$$

and $w(k)$ as $\Delta(p(k))z(k)$, the following LPV state space representation is obtained

$$x(k+1) = \mathcal{A}(p(k))x(k) + \mathcal{B}(p(k))u(k), \quad \text{(3a)}$$
$$y(k) = \mathcal{C}(p(k))x(k) + \mathcal{D}(p(k))u(k) + e(k), \quad \text{(3b)}$$

where

$$\begin{aligned}
\mathcal{A}(p(k)) &= A + B_1\Delta(p(k))(I - D_{11}\Delta(p(k)))^{-1}C_1,\\
\mathcal{B}(p(k)) &= B_2 + B_1\Delta(p(k))(I - D_{11}\Delta(p(k)))^{-1}D_{12},\\
\mathcal{C}(p(k)) &= C_2 + D_{21}\Delta(p(k))(I - D_{11}\Delta(p(k)))^{-1}C_1,\\
\mathcal{D}(p(k)) &= D_{22} + D_{21}\Delta(p(k))(I - D_{11}\Delta(p(k)))^{-1}D_{12}.
\end{aligned}$$

The state sequence $x(k)$ of the LPV model (3) compatible with the date set $\mathcal{D}$ can be estimated by using KCCA as introduced in [11]. The key idea behind using KCCA approach is that the state is the minimal interface between past and future inputs u, outputs y, and scheduling data samples p. Therefore, maximizing the correlation between past and future data samples yields state estimates which are compatible with the data set $\mathcal{D}$. For a given horizon length d, let us define past $\bar{p}_k^d \in \mathbb{R}^{dn_p}$ and future scheduling data vector $\bar{p}_{k+d}^d \in \mathbb{R}^{dn_p}$ w.r.t. time instance k as

$$\bar{p}_k^d = [p(k-d)\ldots p(k-1)]^\top, \bar{p}_{k+d}^d = [p(k)\ldots p(k+d-1)]^\top.$$

The past and future input data $\bar{u}_k^d \in \mathbb{R}^{dn_u}$, $\bar{u}_{k+d}^d \in \mathbb{R}^{dn_u}$ and output data $\bar{y}_k^d \in \mathbb{R}^{dn_y}$, $\bar{y}_{k+d}^d \in \mathbb{R}^{dn_y}$ are defined in a similar manner.

The future output samples of the LPV model (3) can be represented in the observability form

$$\bar{y}_{k+d}^d = (\mathcal{O}_f^d \diamond p)(k)\cdot x(k) + (\mathcal{H}_f^d \diamond p)(k)\cdot \bar{u}_{k+d}^d + \bar{e}_{k+d}^d, \quad (4)$$

where the d-step forward observability matrix $(\mathcal{O}_f^d \diamond p)(k) \in \mathbb{R}^{dn_y\times n_x}$ and the forward Toeplitz matrix $(\mathcal{H}_f^d \diamond p)(k) \in$

$$\underbrace{\begin{bmatrix} \mathcal{C}(p_k) \\ \mathcal{C}(p_{k+1})\mathcal{A}(p_k) \\ \vdots \\ \mathcal{C}(p_{k+d-1})\prod_{j=2}^{d}\mathcal{A}(p_{k+d-j}) \end{bmatrix}}_{(\mathcal{O}_f^d \diamond p)(k)}, \underbrace{\begin{bmatrix} \mathcal{D}(p_k) & 0 & \cdots & 0 \\ \mathcal{C}(p_{k+1})\mathcal{B}(p_k) & \mathcal{D}(p_k) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}(p_{k+d-1})\prod_{j=2}^{d-1}\mathcal{A}(p_{k+d-j})\mathcal{B}(p_k) & \mathcal{C}(p_{k+d-1})\prod_{j=2}^{d-2}\mathcal{A}(p_{k+d-j})\mathcal{B}(p_{k+1}) & \cdots & \mathcal{D}(p_{k+d-1}) \end{bmatrix}}_{(\mathcal{H}_f^d \diamond p)(k)} \tag{5}$$

$\mathbb{R}^{dn_y \times dn_u}$, which have dynamic dependency on p, are explicitly given in (5). From (4), the state variable $x(k)$ is given as[1]

$$\hat{x}(k) = \left((\mathcal{O}_f^d \diamond p)(k)\right)^{\dagger} \left(\bar{y}_{k+d}^d - (\mathcal{H}_f^d \diamond p)(k)\bar{u}_{k+d}^d\right), \tag{6}$$

which shows that the value of the state $x(k)$ at time k is a function of future input, output and scheduling variables. Here, we assume that $\mathcal{O}_f^d$ satisfies the structural observability assumption (see [15, Definition 3.34]) and has full column rank for all $k \in \mathbb{Z}$.

Similarly, using the state update equation (3a), $x(k)$ can be written in terms of past data samples as

$$x(k) = \left(\prod_{i=1}^{d}\mathcal{A}(p_{k-i})\right)x(k-d) + (\mathcal{R}_p^d \diamond p)(k) \cdot \bar{u}_k^d, \tag{7}$$

where $(\mathcal{R}_p^d \diamond p)(k) \in \mathbb{R}^{n_x \times dn_u}$ is the d-step backward reachability matrix which can be easily computed from recursive substitutions of (3a). Here, we also assume that $\mathcal{R}_p^d$ satisfies the structural reachability assumption (see [15, Definition 3.37]).

Shifting (6) by d-samples backward in time we obtain $x(k-d)$ which is then substituted in (7). More specifically, from (6) we have

$$x(k-d) = \left((\mathcal{O}_f^d \diamond p)(k-d)\right)^{\dagger}\left(\bar{y}_k^d - (\mathcal{H}_f^d \diamond p)(k-d)\bar{u}_k^d\right). \tag{8}$$

Then, by substituting (8) in (7),

$$\hat{x}(k) = \mathcal{M}_p^d(k)\left(\bar{y}_k^d - (\mathcal{H}_f^d \diamond p)(k-d)\bar{u}_k^d\right) + (\mathcal{R}_p^d \diamond p)(k)\cdot\bar{u}_k^d, \tag{9}$$

where $\mathcal{M}_p^d(k) = \prod_{i=1}^{d}\mathcal{A}(p_{k-i})\left((\mathcal{O}_f^d \diamond p)(k-d)\right)^{\dagger}$. Thus, from (9) it follows that the conditional expectation of the state variable $x(k)$ at time k is a function of past inputs, outputs and scheduling variable data samples. By defining $\bar{z}_k^d = \begin{bmatrix}\bar{u}_k^d \\ \bar{y}_k^d\end{bmatrix}$, $\bar{z}_{k+d}^d = \begin{bmatrix}\bar{u}_{k+d}^d \\ \bar{y}_{k+d}^d\end{bmatrix} \in \mathbb{R}^{d(n_u+n_y)}$, (6) and (9) can be rewritten as

$$x(k) = \underbrace{\left((\mathcal{O}_f^d \diamond p)(k)\right)^{\dagger}\begin{bmatrix}-(\mathcal{H}_f^d \diamond p)(k) & I\end{bmatrix}}_{\varphi_f(\bar{p}_{k+d}^d)}\bar{z}_{k+d}^d, \tag{10}$$

$$x(k) = \underbrace{\begin{bmatrix}-\mathcal{M}_p^d(k)(\mathcal{H}_f^d \diamond p)(k-d) + (\mathcal{R}_p^d \diamond p)(k) & \mathcal{M}_p^d(k)\end{bmatrix}}_{\varphi_p(\bar{p}_k^d)}\bar{z}_k^d, \tag{11}$$

where $\varphi_p : \mathbb{R}^{dn_p} \to \mathbb{R}^{n_G \times d(n_u+n_y)}$ and $\varphi_f : \mathbb{R}^{dn_p} \to \mathbb{R}^{n_G \times d(n_u+n_y)}$ are (unknown) mappings of past and future scheduling variable samples to an n_G-dimensional feature space. Note that (10) and (11) both provide the representation of the state $x(k)$. However, (10) depends only on future data, while (11) depends only on the past data.

According to the KCCA framework, the state sequence can be estimated by maximizing the correlation between appropriate projections of the maps $\varphi_p(\bar{p}_k^d)\bar{z}_k^d$ and $\varphi_f(\bar{p}_{k+d}^d)\bar{z}_{k+d}^d$. In the following, we present the formulation of the regularized KCCA based on Least-Square Support Vector Machines (LS-SVM) originally introduced in [13] which allows us to estimate the state sequence without explicitly parameterizing the maps $\varphi_p(\bar{p}_k^d)$ and $\varphi_f(\bar{p}_{k+d}^d)$. The regularized version of KCCA formulated with LS-SVM overcomes the problem of "naive kernelization" (see [1]). To develop the LS-SVM formulation we define the $N \times n_G$-dimensional matrices

$$\Phi_p = \left[\varphi_p(\bar{p}_1^d)\bar{z}_1^d,\ \ldots,\ \varphi_p(\bar{p}_N^d)\bar{z}_N^d\right]^{\top}, \tag{12}$$

$$\Phi_f = \left[\varphi_f(\bar{p}_{1+d}^d)\bar{z}_{1+d}^d,\ \ldots,\ \varphi_f(\bar{p}_{N+d}^d)\bar{z}_N^{N+d}\right]^{\top}. \tag{13}$$

The primal LS-SVM problem for KCCA is given by

$$\begin{aligned} &\max_{v_j, w_j} \sum_{k=1}^{N}\left(\gamma s_k r_k - \gamma_f\frac{1}{2}s_k^2 - \gamma_p\frac{1}{2}r_k^2\right) - \frac{1}{2}v_j^{\top}v_j - \frac{1}{2}w_j^{\top}w_j \\ &\text{s.t. } s_k = v_j^{\top}\varphi_f(\bar{p}_{k+d}^d)\bar{z}_{k+d}^d,\ r_k = w_j^{\top}\varphi_p(\bar{p}_k^d)\bar{z}_k^d, \forall k = \mathbb{I}_1^N, \end{aligned} \tag{14}$$

where $\gamma, \gamma_p, \gamma_f \in \mathbb{R}^+$ are regularization hyper-parameters. The variables $v_j \in \mathbb{R}^{n_G}$ and $w_j \in \mathbb{R}^{n_G}$ optimizing (14) represent the directions in the feature space along which the projections of future and past data (s_k and r_k, respectively) have maximum correlation. In order to solve (14) without explicitly specifying the feature maps $\varphi_p(\bar{p}_k^d)$ and $\varphi_f(\bar{p}_{k+d}^d)$, the dual problem is constructed by defining the Lagrangian

$$\begin{aligned} &\mathcal{L}(v_j, w_j, s, r, \eta_j, \kappa_j) = \\ &\sum_{k=1}^{N}\left(\gamma s_k r_k - \gamma_f\frac{1}{2}s_k^2 - \gamma_p\frac{1}{2}r_k^2\right) - \frac{1}{2}v_j^{\top}v_j - \frac{1}{2}w_j^{\top}w_j \\ &- \sum_{k=1}^{N}\eta_j^k\left(s_k - v_j^{\top}\varphi_f(\bar{p}_{k+d}^d)\bar{z}_{k+d}^d\right) \\ &- \sum_{k=1}^{N}\kappa_j^k\left(r_k - w_j^{\top}\varphi_p(\bar{p}_k^d)\bar{z}_k^d\right), \end{aligned} \tag{15}$$

where $\eta_j = \left[\eta_j^1 \cdots \eta_j^N\right]^{\top} \in \mathbb{R}^N$ and $\kappa_j = \left[\kappa_j^1 \cdots \kappa_j^N\right]^{\top} \in \mathbb{R}^N$ are dual Lagrange multipliers. The dual variables η_j and κ_j can be obtained via Karush-Kuhn-Tucker (KKT) conditions, i.e., by setting the gradients of the Lagrangian w.r.t. primal and dual variables $\frac{\partial\mathcal{L}}{\partial v_j}, \frac{\partial\mathcal{L}}{\partial w_j}, \frac{\partial\mathcal{L}}{\partial s_k}, \frac{\partial\mathcal{L}}{\partial r_k}, \frac{\partial\mathcal{L}}{\partial \eta_j^k}, \frac{\partial\mathcal{L}}{\partial \kappa_j^k}$, to zero. Through simple algebraic manipulations, the primal

[1] As e is a zero-mean, independent and identically distributed white noise process, the expected value of the term $\left((\mathcal{O}_f^d \diamond p)(k)\right)^{\dagger}\bar{e}_{k+d}^d$ is zero, which gives the unbiased state estimate (6) in conditional mean sense.

variables can be eliminated based on the KKT conditions. The optimal dual variables need to satisfy the following generalized eigenvalue problem,

$$\begin{bmatrix} 0 & K_{\rm pp} \\ K_{\rm ff} & 0 \end{bmatrix} \begin{bmatrix} \eta_j \\ \kappa_j \end{bmatrix} = \lambda_j \begin{bmatrix} \gamma_f K_{\rm ff} + I & 0 \\ 0 & \gamma_p K_{\rm pp} + I \end{bmatrix} \begin{bmatrix} \eta_j \\ \kappa_j \end{bmatrix}, \quad (16)$$

where $\lambda_j = 1/\gamma$, and $K_{\rm pp} = \Phi_p \Phi_p^\top$ and $K_{\rm ff} = \Phi_f \Phi_f^\top$ are kernel matrices which define the inner product in the feature space. The elements of the kernel matrices are given as

$$[K_{\rm ff}]_{l,m} = (\bar{z}^d_{l+d})^\top \underbrace{\varphi_f^\top(\bar{p}^d_{l+d})\varphi_f(\bar{p}^d_{m+d})}_{\bar{k}(\bar{p}^d_{l+d}, \bar{p}^d_{m+d})} \bar{z}^d_{m+d}, \quad (17a)$$

$$[K_{\rm pp}]_{l,m} = (\bar{z}^d_{l})^\top \underbrace{\varphi_p^\top(\bar{p}^d_{l})\varphi_p(\bar{p}^d_{m})}_{\bar{k}(\bar{p}^d_{l}, \bar{p}^d_{m})} \bar{z}^d_{m}, \quad (17b)$$

In (17), the function $\bar{k}(\cdot,\cdot)$ is a positive definite kernel defining the inner products $\varphi_f^\top(\bar{p}^d_{l+d})\varphi_f(\bar{p}^d_{m+d})$ and $\varphi_p^\top(\bar{p}^d_{l})\varphi_p(\bar{p}^d_{m})$. Definition of the kernel instead of feature maps $\varphi_p(\bar{p}^d_k)$ and $\varphi_f(\bar{p}^d_{k+d})$ is called the *kernel trick* and allows one to formulate the generalized eigenvalue problem (16). An example of the kernel is radial basis function (RBF) $\bar{k}(p_i, p_j) = c \exp\left(-\frac{\|p_i - p_j\|^2}{\sigma^2}\right)$ where c and σ are hyper-parameters which are usually tuned via cross validation.

By solving the generalized eigenvalue problem (16), the dual variables η_j and κ_j are obtained. From the KKT conditions, the primal variables are given as $v_j = \Phi_f^\top \eta_j$ and $w_j = \Phi_p^\top \kappa_j$.

The estimate of state sequence $\hat{x}(k)$ for the LPV model (3) compatible with the data set $\mathcal{D}$ is obtained as follows. We parametrize the j-th component of the estimated state vector $\hat{x}(k)$ as

$$[\hat{x}(k)]_j = v_j^\top \varphi_f(\bar{p}^d_{k+d}) \bar{z}^d_{k+d}$$

Substituting the primal variable $v_j = \Phi_f^\top \eta_j$, and representing the inner product $\varphi_f^\top(\cdot)\varphi_f(\cdot)$ in terms of the kernel function $\bar{k}(\cdot,\cdot)$, we obtain

$$[\hat{x}(k)]_j = \eta_j^\top \begin{bmatrix} \bar{z}^{d\top}_{1+d} \bar{k}(\bar{p}^d_{1+d}, \bar{p}^d_{k+d}) \\ \vdots \\ \bar{z}^{d\top}_{N+d} \bar{k}(\bar{p}^d_{N+d}, \bar{p}^d_{k+d}) \end{bmatrix} \bar{z}^d_{k+d}. \quad (18)$$

Similarly, for $w_j = \Phi_p^\top \kappa_j$ the estimate of the j-th component of the state at time k is given by

$$[\hat{x}(k)]_j = \kappa_j^\top \begin{bmatrix} \bar{z}^{d\top}_{1} \bar{k}(\bar{p}^d_{1}, \bar{p}^d_{k}) \\ \vdots \\ \bar{z}^{d\top}_{N} \bar{k}(\bar{p}^d_{N}, \bar{p}^d_{k}) \end{bmatrix} \bar{z}^d_{k}. \quad (19)$$

$[\hat{x}(k)]_j$ in (18) and (19) are the estimates of the same state value, which are averaged in CCA approach.

Remark: Note that, the eigenvalue problem (16) has N different solutions η_j, κ_j for $j = 1, \ldots, N$. The dual solutions $\eta_j, \kappa_j \in \mathbb{R}^N$ of the generalized eigenvalue problem (16) can be computed using following economical *singular value decomposition* (SVD) [11]

$$\left[\begin{array}{c|c} \gamma_f K_{\rm ff} + I & 0 \\ \hline 0 & \gamma_p K_{\rm pp} + I \end{array}\right]^{-1} \left[\begin{array}{c|c} 0 & K_{\rm pp} \\ \hline K_{\rm ff} & 0 \end{array}\right] = U \Sigma \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}^\top \quad (20)$$

Algorithm 1 Coordinate descent for the estimation of latent variables z and model parameter matrices Θ

Input: training dataset $\mathcal{D} = \{u(k), y(k), p(k)\}_{k=1}^N$; estimated state sequence $\hat{X} = \{\hat{x}(k)\}_{k=1}^N$; tolerance ϵ, maximum number of iteration $n_{\max}$; initial guess Z^0

1. **Iterate for** $n = 1, \ldots$
   1.1 $\Theta^n \leftarrow \operatorname{argmin}_\Theta \quad \mathcal{J}(Z^{n-1}, \Theta)$
   1.2 $Z^n \leftarrow \operatorname{argmin}_Z \quad \mathcal{J}(Z, \Theta^n)$
2. **Until** $\|Z^n - Z^{n-1}\| \leq \epsilon$ or $n = n_{\max}$

Output: Estimated matrices Θ^n

where the dual solutions can be obtained as $\eta_j = [V_1]_j$ and $\kappa_j = [V_2]_j$ with $[\cdot]_j$ denoting the j-th column of the matrix. The dimension of the state $\hat{x}$ can be chosen by considering only those $[\hat{x}(k)]_j$ which correspond to the $\hat{n}_{\rm x}$ most significant singular values contained in Σ. We remark that the estimated state sequence $\hat{x}(k)$, compatible with the data $\mathcal{D}$ is estimated up to a state transformation $T : \mathbb{R}^{dn_p} \to \mathbb{R}^{\hat{n}_{\rm x} \times n_x}$ which can have dynamic dependence on scheduling variable $p(k), \ldots, p(k+d-1)$.

B. Estimation of $z(k)$ and the LTI model parameters

1) General case: Once the state sequence $\hat{X} = \{\hat{x}(k)\}_{k=1}^N$ is estimated, the latent variables $Z = \{z(k)\}_{k=1}^N$ and the unknown matrices $\Theta = \{A, B_1, \ldots, D_{22}\}$ of the LTI model (1) can be obtained (up to a similarity transformation) by minimizing the following non-linear least-squares cost

$$\begin{aligned} \mathcal{J}(Z,\Theta) = & \sum_{k=1}^{N-1} \|\hat{x}(k+1) - (A\hat{x}(k) + B_1 \Delta(p_k) z(k) + B_2 u(k))\|^2_{Q_x} \\ & + \sum_{k=1}^{N} \|z(k) - (C_1 \hat{x}(k) + D_{11} \Delta(p_k) z(k) + D_{12} u(k))\|^2_{Q_z} \\ & + \sum_{k=1}^{N} \|y(k) - (C_2 \hat{x}(k) + D_{21} \Delta(p_k) z(k) + D_{22} u(k))\|^2_{Q_y} \end{aligned} \quad (21)$$

where $Q_x, Q_z, Q_y \succ 0$ are positive definite weighting matrices. The cost function is minimized w.r.t. $\{Z, \Theta\}$ using the coordinate descent approach described in Algorithm 1 under the assumption that $\Delta(p(k))$ is known. Note that, the solution at Step 1.1 and 1.2 of the algorithm can be computed analytically through linear least-squares.

In Algorithm 1, initial guess Z^0 values are chosen randomly from a uniform random distribution.

2) Affine dependence: Although in the general case there is no reliable initialization for Z^0, in the special case of *affine* dependence of (3) on $\Delta(p(k))$, we can use Algorithm 1 in an efficient manner. Hence, consider the LFR model (1) with the following assumptions:

- $D_{11} = 0$, which corresponds to LPV models with affine dependence on $\Delta(p(k))$,
- $\Delta(p(k)) = \varphi(p(k))I$, i.e., the feedback block Δ has a diagonal structure with known real-valued basis functions $\varphi : \mathbb{P} \to \mathbb{R}$.

As $D_{11} = 0$, we have

$$z(k) = C_1 x(k) + D_{12} u(k). \tag{22}$$

By substitutng (22) into (1) we obtain the following state space model,

$$\begin{aligned} x(k+1) =& Ax(k) + B_1 \Delta(p(k)) C_1 x(k) \\ &+ B_1 \Delta(p(k)) D_{12} u(k) + B_2 u(k), \qquad (23a) \\ y(k) =& C_2 x(k) + D_{21} \Delta(p(k)) C_1 x(k) \\ &+ D_{21} \Delta(p(k)) D_{12} u(k) + D_{22} u(k). \qquad (23b) \end{aligned}$$

Using the assumption $\Delta(p(k)) = \varphi(p(k))I$, the unknown matrices Θ in (23), can be solved in a least-squares sense by minimizing the cost

$$\mathcal{J}(\Theta_1, \Theta_2) = \sum_{k=1}^{N-1} \left\| \begin{bmatrix} \hat{x}(k+1) \\ y(k) \end{bmatrix} - \underbrace{\begin{bmatrix} \Theta_1 & \Theta_2 \end{bmatrix}}_{\Theta} \begin{bmatrix} 1 \\ \varphi(p(k)) \end{bmatrix} \otimes \begin{bmatrix} \hat{x}(k) \\ u(k) \end{bmatrix} \right\|_Q^2, \tag{24}$$

where $Q = \mathrm{diag}(Q_x, Q_y) \succ 0$,

$$\Theta_1 = \begin{bmatrix} A & B_2 \\ C_2 & D_{22} \end{bmatrix},$$

$$\Theta_2 = \begin{bmatrix} B_1 C_1 & B_1 D_{12} \\ D_{21} C_1 & D_{21} D_{12} \end{bmatrix} = \begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} \begin{bmatrix} C_1 & D_{12} \end{bmatrix}.$$

Once the least-squares cost $\mathcal{J}$ in (24) is minimized w.r.t. Θ_1 and Θ_2, the matrices $\{A, B_2, C_2, D_{22}\}$ are obtained by appropriate partitioning of Θ_1. The matrices $\{B_1, D_{21}, C_1, D_{12}\}$ are reconstructed by *economic Singular Value Decomposition* (SVD) of Θ_2. Specifically, let U, Σ and V be the matrices obtained from the economic SVD of Θ_2, i.e. $\Theta_2 = U\Sigma V^\top$ which gives

$$\begin{bmatrix} B_1 \\ D_{21} \end{bmatrix} = U\Sigma^{1/2}, \quad \begin{bmatrix} C_1 & D_{12} \end{bmatrix} = \Sigma^{1/2} V^\top. \tag{25}$$

The individual matrices are obtained by proper partitioning using the estimated state dimensions $\hat{n}_\mathrm{x}$ and known dimensions of inputs n_u, outputs n_y and $\Delta(p)$. In this way, the computational complexity is reduced manyfolds as the iterations used in Algorithm 1 are not required.

V. NUMERICAL EXAMPLES

In this section, the effectiveness of the proposed method is demonstrated via simulation examples. The output samples $y(k)$ used in the training phase are corrupted by an additive zero-mean white noise $e(k)$ with Gaussian distribution. The effect of the noise on the output is quantified in terms of the *Signal-to-Noise Ratio* (SNR) defined as $\mathrm{SNR} = 10 \log \frac{\sum_{k=1}^{N} (y(k)-e(k))^2}{\sum_{k=1}^{N} (e(k))^2}$. The quality of the estimated model is assessed on a noise-free validation data of length N_val via *Best Fit Rate* (BFR) and *Variance Accounted For* (VAF) criterion defined as

$$\mathrm{BFR} = \max\left\{1 - \sqrt{\frac{\sum_{k=1}^{N_\mathrm{val}} (y(k) - \hat{y}(k))^2}{\sum_{k=1}^{N_\mathrm{val}} (y(k) - \bar{y})^2}}, 0\right\} \times 100\%,$$

$$\mathrm{VAF} = \max\left\{1 - \frac{\mathrm{var}(y - \hat{y})}{\mathrm{var}(y)}, 0\right\} \times 100\%,$$

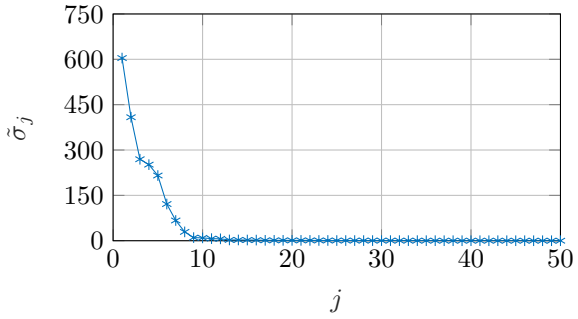

Fig. 2: Singular values of the SVD problem (20)

with $\hat{y}$ being the simulated model output and $\bar{y}$ being the sample mean of the output over the validation set. The operator $\mathrm{var}(\cdot)$ denotes the variance of its argument.

All computations are carried out on i5 1.7GHz Intel core processor with 4 GB of RAM, running MATLAB R2016b.

Example 1

The LPV system (1) is used for data generation with

$$\left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 1.073 \\ -0.1 & 0.7 & 0.816 & 1.075 \\ \hline 0.524 & -0.625 & -0.5 & 0.5 \\ 0.443 & 0.060 & 0.5 & 0.5 \end{array}\right], \tag{26}$$

and the "feedback" path with scalar scheduling variable represented by $w(k) = p(k)z(k)$. Note that, (26) corresponds to rational dependency of (3) on $p(k)$.

Training and validation datasets of length $N = 400$ and $N_\mathrm{val} = 400$, respectively, are generated by exciting system (1) with input u being a white-noise process with uniform distribution $\mathcal{U}(-1, 1)$. The standard deviation of the noise $e(k)$ corrupting the training output $y(k)$ is 0.05, which corresponds to an SNR equal to 21 dB.

The scheduling signal is given by $p(k) = 0.5\sin(k) + \delta(k)$, where $\delta(k)$ is a random variable with uniform distribution $\mathcal{U}(-0.5, 0.5)$. In the first stage, the KCCA algorithm is run to estimate the state sequence $\hat{x}(k)$, using RBF kernels $\bar{k}(p_i, p_j) = c\, \exp\left(-\frac{\|p_i - p_j\|^2}{\sigma^2}\right)$ with parameters $c = 2$ and $\sigma = 10.5$ to construct kernel matrices K_pp and K_ff in (16). The LS-SVM regularization parameters are chosen based on grid search as $\gamma_p = \gamma_f = 500$ and the past and future window length is $d = 3$. These hyper-parameters are chosen through cross-validation. The dimension of the state is determined by solving the SVD problem (20). The first 50 singular values are shown in Fig. 2. We observe that there is a significant gap between the first two singular values and rest of them. Based on this observation the SVD is truncated to the first two components. In other words, the selected state dimension is $n_x = 2$. The state order of $n_x > 2$ can be selected for higher accuracy, but at the cost of larger computation time. The total computation time to construct the state sequence $\hat{x}(k)$ is 8.7 sec. This includes the time required to solve the generalized eigenvalue problem (16) and to obtain $\hat{x}(k)$ based on (18).

Using the estimated state $\hat{x}(k)$, the cost function $\mathcal{J}(Z, \theta)$ in (21) is optimized using a coordinate descent (Algorithm

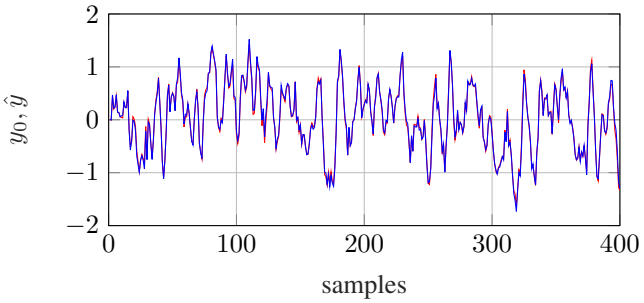

Fig. 3: Example 1. True (red) vs estimated (blue) output

TABLE I: Best Fit Rate (BFR) and Variance Accounted For (VAF) on a noise-free validation data

	Example 1	Example 2
BFR	92.71 %	96.05 %
VAF	99.74 %	99.84 %

1) with $Q_x = 100$, $Q_y = 10$ and $Q_z = 10$, chosen heuristically based on numerical conditions. The initial guess Z^0 is selected from a random distribution. The algorithm is run for $n = 250$ iterations, where each iteration takes around 0.3 sec. The performance of the proposed approach is evaluated using a noise-free validation dataset. The true and the estimated outputs are shown in Fig. 3 and the BFR and VAF criterion are reported in Table I. The obtained results show a good match between system and model output.

Example 2

We consider again the data generating system (26) but now with $D_{11} = 0$ (instead of $D_{11} = -0.5$), which corresponds to an LPV model with affine dependence on $\Delta(p(k))$. The state sequence is first estimated using KCCA with the same hyper-parameters used in the Example 1. Then, the least-square problem (24) is solved followed by singular value decomposition as in (25) to compute the estimates of the unknown model matrices. The total computation time to solve the least squares problem followed by SVD is 1.01 sec. The estimation results in terms of BFR and VAF criterion (computed w.r.t. noise free validation data) are reported in Table I, for an SNR of 21 dB on training data. The results show that an accurate estimate of the output of the true system is obtained with high computational efficiency.

VI. CONCLUSIONS AND FUTURE WORKS

In this contribution, we have proposed a method to identify LPV models in a linear fractional representation. The proposed two stage approach consists first in estimating the state sequence using canonical correlation analysis between past and future data. Then, the latent variables in the LFR form and unknown model parameter matrices are estimated by solving a non-linear least squares problem using a coordinate descent algorithm. The proposed method is applicable for MIMO LPV models defined via LFRs with multi-dimensional scheduling signal. Current research activities are focused on the development of alternative algorithms for latent variable estimation and on the application of the presented KCCA based approach to the identification of other model classes such as switched models.

REFERENCES

[1] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research (JMLR)*, 3:1–49, 2003.

[2] B. A. Bamieh and L. Giarré. Identification of linear parameter-varying models. *International Journal of Robust Nonlinear Control*, 12(9):841–853, 2002.

[3] M. Butcher, A. Karimi, and R. Longchamp. On the consistency of certain identification methods for linear parameter varying systems. In *Proc. of the 17th IFAC World Congress*, pages 4018–4023, Seoul, South Korea, 2008.

[4] Y. Cheng and M. Sznaier. Identification of LPV systems with LFT parametric dependence via convex optimization. In *Proc. of the 54th IEEE Conference on Decision and Control*, pages 1459–1464, Osaka, Japan, 2015.

[5] L. H. Lee and K. Poolla. Identification of linear parameter-varying systems via LFTs. In *Proc. of the 35th IEEE Conference on Decision and Control*, pages 1545–1550, Kobe, Japan, 1996.

[6] L. H. Lee and K. Poolla. Identification of linear parameter-varying systems using nonlinear programming. *Journal of dynamic systems, measurement and Control*, 121(1):71–78, 1999.

[7] M. Mejari, D. Piga, and A. Bemporad. Regularized least square support vector machines for order and structure selection of LPV-ARX models. In *Proc. of the 15th European Control Conference*, pages 1649–1654, Aalborg, Denmark, 2016.

[8] M. Mejari, D. Piga, and A. Bemporad. A bias-correction method for closed-loop identification of Linear Parameter-Varying systems. *Automatica*, 87:128–141, 2018.

[9] M. Nemani, R. Ravikanth, and B. A. Bamieh. Identification of linear parametrically varying systems. In *Proc. of the 34th IEEE Conference on Decision and Control*, pages 2990—-2995, New Orleans, LA, 1995.

[10] D. Piga, P. Cox, R. Tóth, and V. Laurain. LPV system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53:329–338, 2015.

[11] S. Rizvi, J. Mohammadpour, F. Abbasi, and R. Tóth. State-space LPV model identification using kernelized machine learning. *Automatica*, 88:38–47, 2018.

[12] C. W. Scherer. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control for time-varying and linear parametrically-varying systems. *Int. Journal of Robust and Nonlinear Control*, 6(9-10):929–952, 1996.

[13] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific Publishing, Singapore, 2002.

[14] M. Tanelli, D. Ardagna, and M. Lovera. Identification of LPV state space models for autonomic web service systems. *IEEE Transactions on Control Systems Technology*, 19(1):93–103, 2011.

[15] R. Tóth. *Modeling and identification of linear parameter-varying systems*. Lecture Notes in Control and Information Sciences, Vol. 403, Springer, Heidelberg, 2010.

[16] R. Tóth, P. S. C. Heuberger, and P. M. J. Van den Hof. Prediction error identification of LPV systems: present and beyond. In J. Mohammadpour and C. W. Scherer, editors, *Control of Linear Parameter Varying Systems with Applications*, pages 27–60. Springer, 2012.

[17] J. W. van Wingerden and M. Verhaegen. Subspace identification of bilinear and LPV systems for open-and closed-loop data. *Automatica*, 45(2):372–381, 2009.

[18] V. Verdult, J. A. K. Suykens, J. Boets, I. Goethals, and B. De Moor. Least squares support vector machines for kernel CCA in nonlinear state-space identification. In *Proc. of the 16th Intl. symp. math. theory of networks and syst*, Leuven, Belgium, 2004.

[19] V. Verdult and M. Verhaegen. Kernel methods for subspace identification of multivariable LPV and bilinear systems. *Automatica*, 41:1557–1565, 2005.

[20] A. Wills and B. Ninness. System identification of linear parameter varying state- space models. In P. Lopes dos Santos, T. P. Azevedo Perdicoúlis, C. Novara, J. A. Ramos, and D. E. Rivera, editors, *Linear Parameter-Varying System Identification: New Developments and Trends*, chapter 11, pages 295–316. World Scientific, 2011.

[21] K. Zhou and J. C. Doyle. *Essentials of Robust Control*. Prentice-Hall, 1998.